FG

HIG-76-3

6

COMPUTER PROGRAMS FOR SEISMOLOGY: SPECIAL APPLICATIONS TO THE HIGH-GAIN LONG-PERIOD SEISMIC NETWORK

By

DUNCAN M. CHESLEY and EDUARD BERG

MARCH 1976

Prepared for
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
under Contract AFOSR-74-2612
ARPA Order 1827

HAWAII INSTITUTE OF GEOPHYSICS
UNIVERSITY OF HAWAII

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

18 19 REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
AFOSR - TR - 77 - 0257

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER
9

4. TITLE (and Subtitle)
6 Computer Programs for Seismology: Special Applications to the High-Gain Long-Period Seismic Network.

5. TYPE OF REPORT & PERIOD COVERED
Interim rept.

6. PERFORMING ORG. REPORT NUMBER
14 HIG-76-3

7. AUTHOR(s)
10 DUNCAN M. CHESLEY ~~and~~ EDUARD BERG

8. CONTRACT OR GRANT NUMBER(s)
AFOSR 74-2612

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Hawaii Institute of Geophysics, Univ of Hawaii. Honolulu, Hawaii 96822

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
62701E 4F10
1827-21

11. CONTROLLING OFFICE NAME AND ADDRESS
ARPA/NMR
1400 Wilson Blvd.
Arlington VA 22209

11

12. REPORT DATE
March 1976

13. NUMBER OF PAGES
-104-

14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office)
AFOSR/NP
Bolling AFB, Bldg.#410
Wash DC 20332

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

12 76p.

15 AF-AFOSR-2612-74

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

ARPA Order-1827

18. SUPPLEMENTARY NOTES

TECH, OTHER

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Programs. HGLP stations, Filter Rotation Signal extraction, Amplitude determination, Ground motion Beam focusing.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes several IBM 370/158 computer programs that have been applied to long-period seismic data on digital tapes. Included are reading routines for tapes generated by the high-gain long-period (HGLP) stations, rotation of the horizontal components, low-, band-, and high-pass digital filters, correlation (or matched filtering), summation of correlations, beam focusing, and ground motion retrieval.

Results of applications, described elsewhere or in preparation, include (cont'd)

164400

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (cont'd.)

extraction and amplitude determination of Rayleigh waves with signal-to-noise ratios near 1 to 10, precision amplitude and phase calibration of entire seismic systems from transducer to final record, beam forming of matched filtered outputs from the ramdomly spaced HGLP station array, and separation of colocated multiple events with time spacing as short as 150 sec or separation of events widely spaced geographically but arriving nearly simultaneously at a given station.

HIG-76-3

COMPUTER PROGRAMS FOR SEISMOLOGY: SPECIAL APPLICATIONS TO THE HIGH-GAIN LONG-PERIOD SEISMIC NETWORK

By
Duncan M. Chesley and Eduard Berg

March 1976

Prepared for
Air Force Office of Scientific Research
under Contract AFOSR-74-2612
ARPA Order No. 1827

Approved by Director
George P. Woollard

Date: 22 March 1976

Abstract

This report describes several IBM 370/158 computer programs that have been applied to long-period seismic data on digital tapes. Included are reading routines for tapes generated by the high-gain long-period (HGLP) stations, rotation of the horizontal components, low-, band- and high-pass digital filters, correlation (or matched filtering), summation of correlations, beam focusing, and ground motion retrieval.

Results of applications, described elsewhere or in preparation, include extraction and amplitude determination of Rayleigh waves with signal-to-noise ratios near 1 to 10, precision amplitude and phase calibration of entire seismic systems from transducer to final record, beam forming of matched filtered outputs from the randomly spaced HGLP station array, and separation of colocated multiple events with time spacing as short as 150 sec or separation of events widely spaced geographically but arriving nearly simultaneously at a given station.

Table of Contents

Page

Acknowledgments

The authors thank Andrew Lazarewicz and Karl Hinck for their assistance with the computer programming. Special gratitude is extended to Dr. G. H. Sutton for his continued interest and support and for many illuminating discussions.

This research was supported by the Advanced Research Project Agency of the Department of Defense and was monitored by the Air Force Office of Scientific Research under Contract AFOSR 74-2612.

Introduction

The system of programs and subroutines described here has evolved over a number of years of research using digital data tapes from High-Gain Long-Period Seismic stations in studies of seismic surface waves. A prototype was described by Pomeroy _et al_. (1969), and operational systems have been documented in detail in a Lamont-Doherty Geological Observatory report (1971). E. Berg, principal investigator on the project, set the goals, elaborated most of the mathematics, and checked and interpreted the results; D. Chesley performed almost all of the programming. The results presented here were accomplished through close daily interaction between programming and verification of the outputs. Since several Hawaii Institute of Geophysics (HIG) staff members contributed to the effort, the programmer's name for each subroutine has been listed under 'source' in the program descriptions.

Published applications of the programs to detection and amplitude determination of extremely weak Rayleigh waves (Berg, 1974, 1975) include examples of both the type of output obtainable from the programs, and the mathematics involved. An additional application is in obtaining the automated amplitude and phase response of a complete seismometer-recording system from step inputs to the calibration coil (or similar calibration devices), where high accuracy is achieved by summing into an operating system as many individual pulses as desired (to eliminate the ever-present background noise) by the use of correlation and subsequent Fourier analysis. The only parameters required are the seismometer mass, the calibration coil constant (referred to the center of mass if appropriate) and current, and the precise onset time of one reference calibration current. This work (Berg and Chesley, in press) is especially applicable to large arrays and remote-control systems with complicated transmission interfaces to the recording site. Other applications to geophysical data come readily to mind, but will not be discussed here.

Most of the routines are written in FORTRAN and are meant for use on the IBM 370/158 computer. The programs accomplish all data handling, from reading the tape to plotting and tabulating the results of various manipulations.

The programmer intended that the programs and subroutines should interact with a minimum of input from the user. Thus a typical main program supplied by the user will consist of a few initialization statements followed by CALL statements that refer to the subroutines, where all of the analysis is performed. If a user wishes to extract information from his tapes using one of the subroutines listed in this report, he needs only to copy the appropriate sample program from the report; or he may generate his own program to access the subroutines according to his specific needs.

The reader is cautioned against modifying the subroutines. Their interaction is quite complex, and a seemingly inocuous change in one routine could require extensive modifications in other subprograms.

Most of the programs are designed for analysis of seismic velocity output (A-channel) data; exceptions are FINDB and PLOTB. The A-channel format is 1 data point per second, 555 seconds per record, on 556-bit-per-inch magnetic tape, and the B channels are digitized at 1 data point per 5 seconds and 111 data points per record. The header of each record gives start time, station number, number of A channels (3), number of B channels (3), and digitizing interval for A channels (1 second). It is emphasized that modifications allowing use of B channels for correlation and other complicated numerical manipulations, would be difficult. At present the B channels may be read from the tape at any interval (1 data pt/5 sec, 1 pt/min, 1 pt/hr, and so on) and plotted on any scale. The filters may also be applied directly with the appropriate value of DLT.

Finally, this report will not describe the mathematical justifications for the procedures used, since they have been provided elsewhere (Berg, 1974, 1975).

SYSTEM STRUCTURE

This section discusses the interactions that accomplish data transfer between subroutines.

KREAD, a routine written in ASSEMBLOR, reads data from the tape, 555 seconds' worth (1 record, 6006 bytes) at a time. The data pass to READ through the array AREA. READ interprets the data, forming for each record a header array and six data arrays. The header array, HEAD (10), consists of (1) record number, (2) station number (22 = KIP), (3) year, (4) day, (5) hour, (6) minute, and (7) second of the start time of the data in the record, (8) number of A (velocity) channels, (9) number of B (displacement) channels, and finally (10) spacing in seconds between data points in the A channels. HEAD (7) and HEAD (10) are not integers, so the use of the EQUIVALENCE (HEAD (1), IEAD (1)) statement is helpful.

Beginning with the first records, FIND checks the times in HEAD (3-7) against the desired start time until it finds the correct starting record. FIND calculates the correct record number by subtracting the time in HEAD (3-7) from the desired start time, dividing the result by 555 sec/record and skipping backward or forward by the correct number of records, repeating this process until it reaches the record containing the desired start time. This procedure reduces execution time but becomes unworkable if there is a large gap in the data. Refer to the description of READ for details.

FIND takes records one by one from READ through A1-3 in COMMON/SMACK/ and loads the data into CH1-3. At the end of the FIND subroutine, SHRINK corrects CH1-3 by deleting data from the beginning and end of each channel that are outside the desired time range. CH1-3 are then of equal length and contain vertical, NS, and EW velocity data, respectively. HEAD (3-7) contains the time of the first data point, and IXMAX (COMMON/X/) contains the length of each array. At this point, FIND returns control to the main program. The data in CH1-3 must be despiked by the statement CALL DSPYK(0), after which they are ready for any of the manipulations described below.

ROTATION--This calculation translates NS and EW data into radial (parallel to propagation direction) and transverse (perpendicular to radial), modes (both in plane of surface). Data in CH1, CH2, and CH3 are not changed in any way. Instead, two new arrays (CHR and CHT, radial and transverse) are filled. The parameters required by the subroutine ROTATE are PHI, the azimuthal angle (N through E) between N and the direction of the earthquake epicenter from the receiving station (measured in degrees), and G, the gain of channel A3 relative to channel A2 (EW relative to NS).

FILTERING-- Three zero-phase shift filters are available in this system: high-pass, low-pass, and band-pass, where high and low refer to frequency. The user calls HPFILT, LPFILT, and BPFILT to calculate the filter coefficients, which he places in the array W whose length is 2*N+1. The user supplies HAMING, which truncates the filters with a Hamming window by changing the contents of W; and the user supplies N to FLTADJ so that start and stop times supplied to FIND can be modified. DOFILT performs the filtering on one channel at a time and changes the contents of CH1-3, CHR, or CHT. The filtering is done in the time domain by a correlation technique between array W and a data channel the user specifies. This technique deletes N points from each end of the filtered array. DOFILT changes the start time in HEAD (1-10) and the value of IXMAX (if ICHG = 1) so that the correct values of HEAD (10) and IXMAX pass to subsequent programs. ICHG should be set to 1 on the final call to DOFILT.

In addition to calculating filter coefficients, HPFILT, BPFILT, and LPFILT supply through COMMON/FILT/ a string and periods (in seconds) to the plotting subroutine (PLT1). PLT1 uses for these values the labels of the plot.

FOURIER ANALYSIS--If the user copies the data in any of the channel arrays (CH1-3, CHR, CHT) into a complex array whose imaginary part is zero and whose length is an integral power of 2, FASTO (Fast Fourier Transforn) may be used. It replaces the data in the array with the complex Fourier coefficients. The resulting array is symmetric about data point N+1 (Nyquist frequency). The second half of the array will be the complex conjugate of the first. The first data point of the array will contain the mean. AVENUL may be used to zero-mean the data prior to loading the complex array.

PLOTTING--PLT1 is designed to plot a single channel seismogram on a XYNETICS plotter. This subroutine draws the seismogram, labels the axes, and writes a title underneath the plot. The title includes station number and three-letter abbreviation, and the time of the first data point, all from HEAD (1-10). With appropriate arguments, PLT1 labels the rotation angle and gain factor (from COMMON/ANGLE/), as well as the filter type and corner periods (from COMMON/FILT/). SETXYN or SETCC supply letter size (H1, H2) and distance (XOFF, YOFF) from the starting pen position to the origin of the plot (bottom end of Y-axis) through COMMON/PLTPAR/. After one seismogram is plotted (one XYNETICS drawing is generated) PLT1 places the pen above and to the left of the plot. A subsequent call to PLT1 will generate a second trace immediately above the first (if the operator has not moved the pen manually). Thus plots are generated from the bottom upward. To receive the plots in the usual order, the user must plot CH3 first and CH1 last.

PLT1 calls XYNETICS subroutines, which are similar to some CALCOMP plotting routines. An output tape is required and each job generates one file on the tape. DTRK must be specified in the JCL for this output tape even though 800 BPI is used. Please read the XYNETICS plotting instruction booklet for more details (XYNETICS, 1973).

PLT1 can find the maximum and minimum values in the array to be plotted, thus making the plot (without labels) exactly the height specified. PLT1 prints the values it has used. If the station number, HEAD(2), is not in PLT1's list, it prints this number and deletes the station name and number from the label. PLT1 can be used to plot any array, but the horizontal axis is labelled correctly only if the data points are one second apart and the correct start time has been previously placed in HEAD (3-7). PLTB may be used if the data points are an integer multiple of 5 sec interval.

A printer plot of any array may be generated by PRPLT1. There are no labels, except that each line contains the data point number and its value. One line per array element is printed.

The procedures above apply to A-channel (velocity) information. If the user wishes to read B-channel (displacement) data, he must use FINDB. Here he has the option of specifying the spacing in seconds between successive data points in his final CH1-3 arrays. Rotation and filtering may be applied directly. PLOTB should be used.

CORRELATION--Cross-correlation is accomplished by using several subroutines. A short seismogram segment (<20 min) called the reference is correlated with a longer segment (<45 min) called the scan. Much information may be extracted from this process, but only programming considerations will be discussed here.

The user supplies start and stop times for the reference and the scan, as well as an origin time for the earthquake represented in the reference, through a BLOCK DATA subprogram. He must also calculate any filter before calling COREAD. COREAD then reads scan and reference from the input tape, rotates and/or filters the data if desired, and stores the results in scratch files on the disk. Note that if COREAD is to be used, FLTADJ, DOFILT, and ROTATE should not be called in the main program. The start and stop times should be those the user wishes to see in the final output plots. COREAD performs all the manipulations of times that are necessitated by time domain correlations.

The user then calls CORRL8, once for each channel. Each call to CORRL8 reads the data from the appropriate scratch file and plots the reference and then the scan. CORRL8 performs the correlation and plots the coefficients. CORRL8 places the output parameters of interest in COMMON/PRINT/ for later listing by TABLE. Finally, it calculates the sum trace. This array (SUM) contains the point-by-point arithmetic mean of all the correlation coefficients produced by CORRL8. For example, if three channels are correlated, each call to CORRL8 adds to the values already in SUM the coefficients calculated by CORRL8, but divided by three. After the user calls CORRL8 three times, SUM will contain the average of three correlations. COREAD initializes SUM to zero. Because of the nature of PLT1, CH3 should be correlated first and CH1 last.

After correlation, a single CALL PLT1 plots the sum trace and all the timing information is correct. CALL TABLE tabulates the results for all channels and CALL WRTSUM writes the sum trace coefficients on disk for later use in beam focusing. At the end of all plotting jobs the statement

```
CALL PLOT (0, HYT, 999)
```

should be used. This places an EOF on the output tape and generates a table showing how many XYNETICS drawings have been produced (1 for every CALL PLT1 plus an extra for the EOF), thus checking the correct operation of the program.

BEAM FOCUSING--A further increase in the signal-to-noise ratio for correlation SUM traces may be accomplished by averaging several SUM traces that were created by calls to WRTSUM. This procedure is accomplished by the program SUMAVE, listed in the following section.

GROUND MOTION--Use of the subroutine INFILT with the other filtering subroutines (FLTADJ, DOFILT) allows retrieval of the ground motion by deconvolution of the seismometer and recording system response from the taped seismic data. The response must be stored on disk. The sample program FILTER1 illustrates this procedure.

References

Berg, Eduard, 1974, Rayleigh Waves from High-Gain Long-Period Stations: Signal Extraction; Amplitude Determination; and Separation of Overlapping Wave Trains, Hawaii Inst. Geophys. Report HIG-74-10, 29 pp., 21 computer outputs.

Berg, Eduard, 1975, Rayleigh Waves from High-Gain Long-Period Stations: Signal Extraction; Amplitude Determination; and Separation of Overlapping Wave Trains, Bull. Seismol. Soc. Am., v. 65, no. 6, p. 1761-1778.

Berg, Eduard, and Duncan M. Chesley, Automated High-Precision Amplitude and Phase Calibration of Seismic Systems, Bull. Seismol. Soc. Am., in press.

Lamont-Doherty Geological Observatory, 1971, High-Gain Long-Period Seismograph Station Instrumentation, Vol. 1 and 2, Advanced Research Project Agency Order 1513, 58 pp.

Pomeroy, R. W., G. Hade, J. Savino, and R. Chander, 1969, Preliminary Results High-Gain Wide-Band Long-Period Electromagnetic Seismograph Systems, J. Geophys. Res., v. 74, p. 3295-3298.

XYNETICS, 1973, XYNETICS Basic Software Fast Mode, XYNETICS, Inc., 6710 Variel Ave., Canoga Park, Ca.,

SUBROUTINES

This section contains lists of all the subroutines in the system. There are four types of subroutines:

FILTER: These subroutines calculate filter coefficients and start times, and are used to apply the various filters to the data. Filtering is accomplished using a correlation technique and requires several routines.

UTILITY: These subroutines perform 'bookkeeping'. They usually are short and usually are not found in main programs.

INPUT-OUTPUT (IO): These read tapes, plot, write files, or print results.

DATA: These programs perform the mathematical manipulations, except for filtering.

The following list should be useful for finding the exact order of the arguments of the subroutines.

```
AVENUL (A, I)

BPFILT (W, N, DLT, BPFREQ, HPFREQ)

COREAD (LREF, PHI, GA)

CORRL8 (ICHAN, SCALE, HYT, LAB, NUMCH)

DOFILT (A, W, N, ICHG)

DSPYK (NUM)

FASTO (A, M, MODE)

FIND (ISTART, ISTOP)

FINDB (ISTART, ISTOP, INTER)

FLTADJ (IL, ISRT, ISTP)

HAMING (W, N)

HDCONV (IHEAD, NUM)

HD2SEC (IA, IB)

HPFILT (W, N, DLT, HPFREQ)

INFILT (ICH, W, N, N1, N2)

KREAD (AREA, NBYTES) or KRDBK (AREA(LAST), NBYTES)

LPFILT (W, N, DLT, BPFREQ)

MAXMIN (A, N, AMAX, MAXJ, AMIN, MINJ)

PLOTB (Z, SCALE, HITE, ZMAX, ZMIN, LKZ, ISPIK, LAB, INTER)

PLT1 (Z, SCALE, HITE, ZMAX, ZMIN, LKZ, ISPIK, LAB)

PRPLT1 (X, LB, LE, LS)

READ (NREC, IPRT)

ROTATE (PHI, G, NUM)

SEC2HD (IB, IA)

SETCC

SETNP
```

SETXYN

SHRINK (IBEG, IEND)

TABLE

WRTSUM (IF)

NAME--AVENUL

TYPE--DATA

SOURCE--MAREK FRYDRICH

PURPOSE and COMMENTS--subtracts the mean from an array of data

DESCRIPTION--CALL AVENUL (A, I)

where: A is the name of the array

I is the length of the array.

```
      SUBRCUTINE AVENUL(A.IXMAX)
      DIMENSION A(1)
      SUM=0.0
      DC 5 I=1.IXMAX
      SUM=SUM+A(I)
    5 CONTINUE
      DO 7 I=1,IXMAX
      A(I)=A(I)-(SUM/IXMAX)
    7 CONTINUE
      RETURN
      END
```

NAME--BPFILT

TYPE--FILTER

SOURCE--D. CHESLEY

PURPOSE--Generates filter coefficients for a band-pass filter. The filter will be applied by a correlation technique (DOFILT) in the time domain.

DESCRIPTION--CALL BPFILT (W, N, DLT, BPFREQ, HPFREQ)

where W - the array for the coefficients

N - the number such that 2*N+1 is the number of coefficients (≤1000)

DLT - the time interval between data points in seconds

BPFREQ - low pass corner frequency (HZ)

HPFREQ - high pass corner frequency (HZ)

This program must be used in conjunction with FLTADJ, DOFILT, and HAMING.

COMMON--/FILT/ is used by PLT1 for correct labelling of the plot.

```
ISN 0002             SUBROUTINE BPFILT(W.N.DLT.BPFREQ.HPFREQ)
              C
              C       W IS ARRAY OF 2*N+1 POINTS
              C       N IS NUMBER OF DATA POINTS EACH SIDE OF MIDDLE
              C
ISN 0003             COMMON/FILT/FILAB(2).SPER.XLPER
ISN 0004             DIMENSION XLAB(2)
ISN 0005             DIMENSION W(1)
ISN 0006             DATA XLAB/'BP F'.'ILT '/
ISN 0007             SPER=1./HPFREQ
ISN 0008             XLPER=1./BPFREQ
ISN 0009             FILAB(1)=XLAB(1)
ISN 0010             FILAB(2)=XLAB(2)
ISN 0011             PI=3.141592653б
ISN 0012             CON1=DLT*BPFREQ*2.
ISN 0013             CON2=DLT*HPFREQ*2.
ISN 0014             DO 10 I=1.N
ISN 0015             X1=(-N+I-1)*PI*CON1
ISN 0016             X2=(-N+I-1)*PI*CON2
ISN 0017             W(I)=CON1*SIN(X1)/X1-CON2*SIN(X2)/X2
ISN 0018             W(2*N-I+2)=W(I)
ISN 0019          10 CONTINUE
ISN 0020             W(N+1)=CON1-CON2
ISN 0021             RETURN
ISN 0022             END
```

NAME--COREAD

TYPE--IO

SOURCE--D. CHESLEY

PURPOSE--COREAD reads data from tape and creates scratch files for subsequent access by CORRL8.

DESCRIPTION--CALL COREAD (LREF, PHI, GA)

LREF - If LREF $\neq$ 0 the program applies a filter to all data.

PHI - angle of rotation

GA - gain factor for rotation

PHI and GA are the arguments of ROTATE. If they are both zero no rotation is performed.

COMMON--/SEARCH/ - contains data that is read off tape

/COR/ - contains array for sum (average correlation) trace and times for reference, origin, and scan

/TIME/ - contains lengths of various arrays, start times of reference and scan, and JUMP, which indicates the order of the reference and scan in time

/ANGLE/ - contains data for PLT labels

/TURN/ - contains rotated data

/ARRAY/ - contains filter length and coefficients

/PRINT/ - contains data for TABLE that are initialized in COREAD

NOTES--CORRL8 operates on only one channel at a time. To avoid rereading the tape for other channels, COREAD reads the data once and creates scratch files, which are then read by CORRL8. This saves execution time and core requirements but adds three more cards to the JCL stream. Suitable modifications to COREAD enable the user to use other types of reference and scan data. Files may be created by an independent main program and suitable parameters supplied to CORRL8 through COMMON. Similar modifications have been used to allow an average calibration pulse as a reference instead of actual data read from magnetic tape by COREAD. The danger in modifications such as these is that values in COMMON areas may be wrong (such as start times or array sizes); these values should be carefully examined in the event of a program failure. The necessary times are supplied to COREAD from a BLOCK DATA subprogram through COMMON/COR/. IREF1 and IREF2 (dimension = 5, integer) contain year, day, hour, minute, and second of the start and stop times of the reference to be read off tape and used by PLT1 labels. IORIG contains the origin time of the reference earthquake in the same format. ISCAN1 and ISCAN2 contain start and stop times for the scan trace.

The user must supply the JCL for defining the scratch files. See sample programs.

```
ISN 0002                SUBROUTINE COREAD(LREF,PHI,GA)
ISN 0003                COMMON/SEARCH/CH1(5550),CH2(5550),CH3(5550)
ISN 0004                COMMON/COR/SUM(2700),IREF1(5),IREF2(5),IORIG(5),
                       .ISCAN1(5),ISCAN2(5)
ISN 0005                COMMON/TIME/NSEC,REFSEC,RFSTRT,SCNSRT,JUMP,ENDSEC
ISN 0006                COMMON/ANGLE/ANG,GAIN
ISN 0007                COMMON/TURN/CHR(5550),CHT(5550)
ISN 0008                COMMON/ARRAY/N,W(1000)
ISN 0009                COMMON/PRINT/JMAX(4),JMIN(4),CMAX(4),CMIN(4),SLOPE1(4),
                       .SLOPE2(4),ERROR1(4),ERROR2(4),AMPL1(4),AMPL2(4),AMPER1(4),
                       .AMPER2(4)
               C
ISN 0010                INTEGER REFSEC,RFSTRT,RFSTOP,SCNSRT,SCNSTP,BEGIN,REFHED
ISN 0011                INTEGER SCN1,SCNHED,SCNCHK,END,TSTART,XS,ENDSEC
ISN 0012                DIMENSION IEAD(10),IHEAD(5),SCN1(5),IR1(5),IR2(5),IS1(5),IS2(5
               C
ISN 0013                DO 20 J=1,2700
ISN 0014         20     SUM(J)=0.0
ISN 0015                JUMP=1
ISN 0016                ANG=PHI
ISN 0017                GAIN=GA
ISN 0018                DO 50 ICHAN=1,4
ISN 0019                JMAX(ICHAN)=0
ISN 0020                JMIN(ICHAN)=0
ISN 0021                CMAX(ICHAN)=-99999.
ISN 0022                CMIN(ICHAN)=99999.
ISN 0023                SLOPE1(ICHAN)=0.
ISN 0024                SLOPE2(ICHAN)=0.
ISN 0025                ERROR1(ICHAN)=0.
ISN 0026                ERROR2(ICHAN)=0.
ISN 0027                AMPL1(ICHAN)=0.
ISN 0028                AMPL2(ICHAN)=0.
ISN 0029                AMPER1(ICHAN)=0.
ISN 0030                AMPER2(ICHAN)=0.
ISN 0031         50     CONTINUE
               C
ISN 0032                DO 70 J=1,5
ISN 0033                IR1(J)=IREF1(J)
ISN 0034                IR2(J)=IREF2(J)
ISN 0035                IS1(J)=ISCAN1(J)
ISN 0036                IS2(J)=ISCAN2(J)
ISN 0037         70     CONTINUE
ISN 0038                IF(LREF.EQ.0)GO TO 90
               C
ISN 0040         71     CONTINUE
ISN 0041                CALL FLTADJ(N,IR1,IR2)
ISN 0042                CALL FLTADJ(N,IS1,IS2)
ISN 0043         72     CONTINUE
ISN 0044                CALL HD2SEC(IR1,IR1SEC)
ISN 0045                CALL HD2SEC(IR2,IR2SEC)
ISN 0046                IRSEC=IR2SEC-IR1SEC+1
ISN 0047                IF(IRSEC.LE.1510) GO TO 80
ISN 0049                XS=IRSEC-1510
ISN 0050                PRINT 10,XS
ISN 0051                GO TO 440
               C
ISN 0052         80     CALL HD2SEC(IS1,IS1SEC)
ISN 0053                CALL HD2SEC(IS2,IS2SEC)
ISN 0054                ISSEC=IS2SEC-IS1SEC+1
               C
ISN 0055         90     CALL HD2SEC(IREF1,RFSTRT)
ISN 0056                CALL HD2SEC(IREF2,RFSTOP)
ISN 0057                REFSEC=RFSTOP-RFSTRT+1
ISN 0058                IF(REFSEC.LE.1510)GO TO 100
ISN 0060                XS=REFSEC-1510
ISN 0061                PRINT 10,XS
ISN 0062         10     FORMAT(1X,'LENGTH OF REFERENCE EXCEEDS DIMENSION BY ',I7)
ISN 0063                STOP
ISN 0064        100     CALL HD2SEC(ISCAN1,SCNSRT)
ISN 0065                CALL HD2SEC(ISCAN2,SCNSTP)
ISN 0066                NSEC=SCNSTP-SCNSRT+REFSEC
ISN 0067                ENDSEC=SCNSTP-SCNSRT+1
ISN 0068                IF(LREF.EQ.0) ISSEC=NSEC
ISN 0070                IF(LREF.NE.0) ISSEC=ISSEC+REFSEC-1
ISN 0072                IF(ISSEC.LE.4210)GO TO 110
ISN 0074              XS=ISSEC-4210
ISN 0075              PRINT 11,XS
ISN 0076        11    FORMAT(1X,'LENGTH OF SCAN EXCEEDS DIMENSION BY ',I7)
ISN 0077              STOP
               C
               C       GET REFERENCE FROM TAPE
               C
ISN 0078        110   IF (SCNSRT.LT.RFSTRT) JUMP=2
ISN 0080              GO TO (111,200),JUMP
ISN 0081        111   CALL FIND(IR1,IR2)
ISN 0082              CALL DSPYK(0)
ISN 0083              IF(GAIN.NE.0.) CALL ROTATE(PHI,GAIN,0)
ISN 0085              IF(LREF.EQ.0) GO TO 120
ISN 0087              CALL DOFILT(CH1,W,N,0)
ISN 0088              IF(GAIN.NE.0.) GO TO 115
ISN 0090              CALL DOFILT(CH2,W,N,0)
ISN 0091              CALL DOFILT(CH3,W,N,1)
ISN 0092              GO TO 120
ISN 0093        115   CALL DOFILT(CHR,W,N,0)
ISN 0094              CALL DOFILT(CHT,W,N,1)
               C
ISN 0095        120   WRITE(11) CH1
ISN 0096              KUMP=2
ISN 0097              IF(GAIN.NE.0.) GO TO 130
ISN 0099              WRITE(12) CH2
ISN 0100              WRITE(13) CH3
ISN 0101              GO TO (200,350),JUMP
ISN 0102        130   WRITE(12) CHR
ISN 0103              WRITE(13) CHT
ISN 0104              GO TO (200,350),JUMP
               C
ISN 0105        200   CALL HD2SEC(IS1,IS1SEC)
ISN 0106              LEN=IS1SEC+ISSEC-1
ISN 0107              CALL SEC2HD(LEN,IS2)
ISN 0108              CALL FIND(IS1,IS2)
ISN 0109              CALL DSPYK(0)
ISN 0110              IF(GAIN.NE.0.) CALL ROTATE(PHI,GAIN,0)
ISN 0112              IF(LREF.EQ.0) GO TO 220
ISN 0114              CALL DOFILT(CH1,W,N,0)
ISN 0115              IF(GAIN.NE.0.) GO TO 210
ISN 0117              CALL DOFILT(CH2,W,N,0)
ISN 0118              CALL DOFILT(CH3,W,N,1)
ISN 0119              GO TO 220
ISN 0120        210   CALL DOFILT(CHR,W,N,0)
ISN 0121              CALL DOFILT(CHT,W,N,1)
               C
ISN 0122        220   WRITE(11) CH1
ISN 0123              IF(GAIN.NE.0.) GO TO 330
ISN 0125               WRITE(12) CH2
ISN 0126              WRITE(13) CH3
ISN 0127              GO TO (350,111),JUMP
ISN 0128        330   WRITE(12) CHR
ISN 0129              WRITE(13) CHT
ISN 0130              GO TO (350,111),JUMP
               C
ISN 0131        350   CONTINUE
ISN 0132              ENDFILE 11
ISN 0133              REWIND 11
ISN 0134              ENDFILE 12
ISN 0135              REWIND 12
ISN 0136              ENDFILE 13
ISN 0137              REWIND 13
ISN 0138        440   RETURN
ISN 0139              END
```

NAME--CORRL8

TYPE--DATA

SOURCE--D. CHESLEY

PURPOSE-- cross correlation of one array, called the reference, with another, called the scan. Operates on one channel. Successive calls create an average trace that is the arithmetic mean, point by point, of all the calculated correlation coefficients. Each call generates three plots: reference, scan, and correlation coefficients, in that order.

DESCRIPTION-- CALL CORRL8 (ICHAN, SCALE, HYT, LAB, NUMCH)

ICHAN - channel number to be correlated. 1 = CH1, 2 = CH2 or CHR, 3 = CH3 or CHT

SCALE - horizontal scale of plotted output cm/min

HYT - height of plot in inches (from minimum to maximum, not including label)

LAB - plotting label parameters (see PLT1)

= 0 no rotation or filter labels

= 1 rotation, no filter

= 2 filter, no rotation

= 3 rotation and filter labels

NUMCH - number of channels that are combined in sum (average correlation) trace (one CALL CORRL8 for each channel)

COMMON - /SMACK/ contains header information for PLT1 labels

/SEARCH/ contains data and correlation coefficients

/X/ contains lengths of plotted arrays

/COR/ contains sum trace and reference, origin, and scan times

/ANGLE/ contains rotation information for PLT1 labels

/TURN/ contains rotated data

/PRINT/ contains output for TABLE

/TIME/ contains lengths of arrays and JUMP, which gives order of scan and reference in time

NOTES--CORRL8 reads files that have been created by COREAD (or a similar process). There is JCL interaction here. Channel 1 data must be stored in file 11, 2 in 12, and so on (see sample program). CORRL8 plots the reference and then the scan. For correct timing information, the data must be one point per second. CORRL8 performs the correlation, plots the coefficients and calculates correlation amplitudes. CORRL8 divides each coefficient by NUMCH and adds the coefficients to the values in SUM. This results in an average correlation after all correlations are finished. (There must be NUMCH calls to CORRL8). CORRL8 passes the output to TABLE for printing. The sum trace must be plotted by the main program.

```
ISN 0002                SUBROUTINE CORRL8(ICHAN,SCALE,HYT,LAB,NUMCH)
ISN 0003                COMMON/SMACK/A1(555),A2(555),A3(555),B1(111),B2(111),
                       .B3(111),HEAD(10),IREC
ISN 0004                COMMON /SEARCH/CH1(5550),CH2(5550),CH3(5550)
ISN 0005                COMMON/X/IXMAX
ISN 0006                COMMON/COR/SUM(2700),IREF1(5),IREF2(5),IORIG(5),
                       .ISCAN1(5),ISCAN2(5)
ISN 0C07                COMMON/ANGLE/ANG,GAIN
ISN 0008                COMMON/TURN/CHR(5550),CHT(5550)
ISN 0009                COMMON/PRINT/JMAX(4),JMIN(4),CMAX(4),CMIN(4),SLOPE1(4),
                       .SLOPE2(4),ERROR1(4),ERROR2(4),AMPL1(4),AMPL2(4),AMPER1(4),
                       .AMPER2(4)
ISN 0010                COMMON/TIME/NSEC,REFSEC,RFSTRT,SCNSRT,JUMP,ENDSEC
ISN 0011                DIMENSIUN IEAD(10),IHEAD(5)
ISN 0012                INTEGER ENDSEC
ISN 0013                INTEGER REFSEC,RFSTRT,SCNSRT,TSTART
ISN 0014                DOUBLE PRECISION SUMX,SUMY,SUMX2,SUMY2,SUMXY,D1,D2,D3
ISN 0015                EQUIVALENCE (HEAD(1),IEAD(1))
                  C
ISN 0016                IF(LAB-1)5,5,6
ISN 0017            5   LABE=0
ISN 0018                GO TO 7
ISN 0019            6   LABE=2
ISN 0020            7   JCHAN=ICHAN+10
ISN 0021                IF(JUMP.EQ.2) GO TO 100
ISN 0023                READ(JCHAN) CH1
ISN 0024                READ(JCHAN) CH2
ISN 0025                REWIND JCHAN
ISN 0026                GO TO 125
ISN 0027          100   READ(JCHAN) CH2
ISN 0028                READ(JCHAN) CH1
ISN 0029                REWIND JCHAN
                  C
                  C      PLOT REFERENCE
                  C
ISN 0030          125   CALL HDCONV(IREF1,-1)
ISN 0031                IXMAX=REFSEC
ISN 0032                PRINT 130,ICHAN
ISN 0033          130   FORMAT(1X,/,'  REFERENCE; CHANNEL A',I1,/)
ISN 0034                CALL MAXMIN(CH1,IXMAX,REFMAX,MAXJ,PEFMIN,MINJ)
ISN 0035                GO TO(133,135,135),ICHAN
ISN 0036          133   CALL PLT1(CH1,SCALE,HYT,0.,0.,'A1',0,LABE)
ISN 0037                GO TO 200
ISN 0039          135   IF(GAIN.NE.0.) GO TO 138
ISN 0040                GO TO(133,136,137),ICHAN
ISN 0041          136   CALL PLT1(CH1,SCALE,HYT,0.,0.,'A2',0,LABE)
ISN 0042                GO TO 200
ISN 0043          137   CALL PLT1(CH1,SCALE,HYT,0.,0.,'A3',0,LARE)
ISN 0044                GO TO 200
ISN 0045          138   GO TO (133,139,140),ICHAN
ISN 0046          139   CALL PLT1(CH1,SCALE,HYT,0.,0.,'AR',0,LAB)
ISN 0047                GO TO 200
ISN 0048          140   CALL PLT1(CH1,SCALE,HYT,0.,0.,'AT',0,LAB)
ISN 0049                GO TO 200
                  C
                  C
                  C      CONVERT HEADER TO DESIRED START TIME
                  C
ISN 0050          200   CALL SEC2HD(SCNSRT,IHEAD)
ISN 0051                CALL HDCONV(IHEAD,-1)
                  C
                  C      PLOT SCAN
                  C
```

```
ISN 0052            IXMAX=NSEC
ISN 0053            PRINT 275.ICHAN
ISN 0054      275   FORMAT(1X./.'   SCAN: CHANNEL A'.I1./)
ISN 0055            GO TO (280.281.281).ICHAN
ISN 0056      280   CALL PLT1(CH2.SCALE.HYT.0..0..'A1'.0.LABE)
ISN 0057            GO TO 300
ISN 0058      281   IF(GAIN.NE.0.) GO TO 285
ISN 0060            GO TO (280.282.283).ICHAN
ISN 0061      282   CALL PLT1(CH2.SCALE.HYT.0..0..'A2'.0.LABE)
ISN 0062            GO TO 300
ISN 0063      283   CALL PLT1(CH2.SCALE.HYT.0..0..'A3'.0.LABE)
ISN 0064            GO TO 300
ISN 0065      285   GO TO(280.286.287).ICHAN
ISN 0066      286   CALL PLT1(CH2.SCALE.HYT.0..0..'AR'.0.LAB)
ISN 0067            GO TO 300
ISN 0068      287   CALL PLT1(CH2.SCALE.HYT.0..0..'AT'.0.LAB)
               C
               C     CONVERT HEADER TO DESIRED START TIME
               C
ISN 0069      300   CALL HD2SEC(IORIG.ITIME)
ISN 0070            TSTART=SCNSRT-RFSTRT+ITIME
ISN 0071            CALL SEC2HD(TSTART.IHEAD)
ISN 0072            CALL HDCONV(IHEAD.-1)
               C
               C     *** BEGIN CORRELATION ROUTINE ***
               C
               C     FIRST CALCULATE CONSTANT TERM
               C
ISN 0073            SUMX2=0.0
ISN 0074            SUMX=0.0
ISN 0075            DO 325 I=1.REFSEC
ISN 0076            SUMX2=SUMX2+CH1(I)*CH1(I)
ISN 0077            SUMX=SUMX+CH1(I)
ISN 0078      325   CONTINUE
               C
ISN 0079            D2=REFSEC*SUMX2-SUMX**2
               C
               C     NOW CALCULATE OTHER TERMS
               C
ISN 0080            SUMY=0.0
ISN 0081            SUMY2=0.0
ISN 0082      326   CONTINUE
ISN 0083            DO 400 J=1.ENDSEC
ISN 0084            SUMXY=0.0
ISN 0085            DO 375 I=1.REFSEC
ISN 0086            IF(J.NE.1) GO TO 370
ISN 0088            SUMY=SUMY+CH2(I+J-1)
ISN 0089            SUMY2=SUMY2+CH2(I+J-1)*CH2(I+J-1)
ISN 0090      370   SUMXY=SUMXY+CH1(I)*CH2(I+J-1)
ISN 0091      375   CONTINUE
               C
ISN 0092            IF(J.EQ.1) GO TO 380
ISN 0094            SUMY=SUMY-SUMYS+CH2(REFSEC+J-1)
ISN 0095            SUMY2=SUMY2-SUMY2S+CH2(REFSEC+J-1)*CH2(REFSEC+J-1)
ISN 0096      380   SUMYS=CH2(J)
ISN 0097            SUMY2S=CH2(J)*CH2(J)
               C
               C     CORRELATION COEFFICIENT
               C
```

```
ISN 0098                D1=REFSEC*SUMXY-SUMX*SUMY
ISN 0099                D3=REFSEC*SUMY2-SUMY**2
ISN 0100                CH2(J)=D1/DSQRT(D2*D3)
ISN 0101                SUM(J)=SUM(J)+CH2(J)/NUMCH
                  C
                  C      CHECK FOR MAXIMUM AND MINIMUM AND CALCULATE
                  C      ERROR AND SLOPE IF NECESSARY
                  C
ISN 0102                IF(CH2(J).LE.CMAX(ICHAN)) GO TO 390
                  C
ISN 0104                JMAX(ICHAN)=J+TSTART-1
ISN 0105                CMAX(ICHAN)=CH2(J)
ISN 0106                SLOPE1(ICHAN)=D1/D2
ISN 0107                ERROR1(ICHAN)=DSQRT((D3/D2-SLOPE1(ICHAN)**2)/(REFSEC-2))
                  C
ISN 0108           390  IF(CH2(J).GE.CMIN(ICHAN)) GO TO 400
ISN 0110                JMIN(ICHAN)=J+TSTART-1
ISN 0111                CMIN(ICHAN)=CH2(J)
ISN 0112                SLOPE2(ICHAN)=D1/D2
ISN 0113                ERROR2(ICHAN)=DSQRT((D3/D2-SLOPE2(ICHAN)**2)/(REFSEC-2))
                  C
ISN 0114           400  CONTINUE
ISN 0115           401  CONTINUE
ISN 0116                DIF=REFMAX-REFMIN
ISN 0117                AMPL1(ICHAN)=DIF*SLOPE1(ICHAN)
ISN 0118                AMPL2(ICHAN)=DIF*SLOPE2(ICHAN)
ISN 0119                AMPER1(ICHAN)=DIF*ERROR1(ICHAN)
ISN 0120                AMPER2(ICHAN)=DIF*ERROR2(ICHAN)
ISN 0121           402  CONTINUE
                  C
                  C      PLOT RESULTS
                  C
ISN 0122                IXMAX=ENDSEC
ISN 0123                GO TO (410,420,420),ICHAN
ISN 0124           410  CALL PLT1(CH2,SCALE,HYT,1.,-1.,'C1',0,0)
ISN 0125                GO TO 440
ISN 0126           420  IF(GAIN.NE.0.)GO TO 430
ISN 0128                GO TO (410,425,426),ICHAN
ISN 0129           425  CALL PLT1(CH2,SCALE,HYT,1.,-1.,'C2',0,0)
ISN 0130                GO TO 440
ISN 0131           426  CALL PLT1(CH2,SCALE,HYT,1.,-1.,'C3',0,0)
ISN 0132                GO TO 440
                  C
ISN 0133           430  GO TO (410,435,436),ICHAN
ISN 0134           435  CALL PLT1(CH2,SCALE,HYT,1.,-1.,'CR',0,0)
ISN 0135                GO TO 440
ISN 0136           436  CALL PLT1(CH2,SCALE,HYT,1.,-1.,'CT',0,0)
                  C
ISN 0137           440  RETURN
ISN 0138                END
```

NAME--DOFILT

TYPE--FILTER

SOURCE--D. CHESLEY

PURPOSE--FILTERS an array of time series data using coefficients calculated by BPFILT, HPFILT, LPFILT and HAMING. DOFILT filters with a correlation technique in time domain.

DESCRIPTION--CALL DOFILT (A, W, N, ICHG)

A - the array to be filtered and the array after filtering

W - array containing the filter coefficients

N - 2*N+1 is the number of coefficients (≤1000)

ICHG - The correlation technique used in DOFILT changes the length of the array after the correlation by dropping N points from each end of the array. ICHG ≠0 changes IXMAX and the start time that goes with the array A.

COMMON--/SMACK/ contains HEAD(10), which contains the start time of the data

/X/ contains IXMAX, which is the length of A. Both these quantities are changed by DOFILT if ICHG≠0.

NOTES--If the user wants to filter all three A channels, then the value of IXMAX and the start time in HEAD(10) should be changed only once. This should be done on the last call to DOFILT as below:

```
CALL HPFILT (W, N, DLT, HPFREQ)

CALL HAMING (W, N)

CALL DOFILT (CH1, W, N, 0)

CALL DOFILT (CH2, W, N, 0)

CALL DOFILT (CH3, W, N, 1).
```

```
          SUBROUTINE DOFILT(A,W,N,ICHG)
          COMMON/SMACK/A1(555),A2(556).A3(555).B1(111).B2(111).
         .B3(111),HEAD(10),IREC
          COMMON/X/IXMAX
          DIMENSION A(1).W(1).IHEAD(5),IEAD(10)
          EQUIVALENCE (HEAD(1).IEAD(1))
          LCHG=ICHG
          IN=1
          N1=N+1
          N2=IXMAX-N
          I1=2*N+1
          DO 20 J=N1,N2
          IMIN=J-N
          COEF=0.
          DO 15 I=1,I1
          COEF=COEF+W(I)*A(IMIN+I-1)
   15     CONTINUE
          A(J-N)=COEF
   20     CONTINUE
          IF(LCHG) 22,999,22
   22     IXMAX=IXMAX-2*N
          CALL HDCONV(IHEAD,1)
          CALL HD2SEC(IHEAD,IS)
          IS=IS+N*IN
          CALL SEC2HD(IS,IHEAD)
          DO 40 J=1,4
   40     IEAD(J+2)=IHEAD(J)
          HEAD(7)=IHEAD(5)
  999     RETURN
          END
```

NAME--DSPYK

TYPE--UTILITY

SOURCE--D. CHESLEY

PURPOSE--Removes spikes from data, three A channels simultaneously.

DESCRIPTION--CALL DSPYK (NUM)

NUM = 0 is used if the data has been read using FIND; NUM = 1 if READ has been used to get the data from the tape.

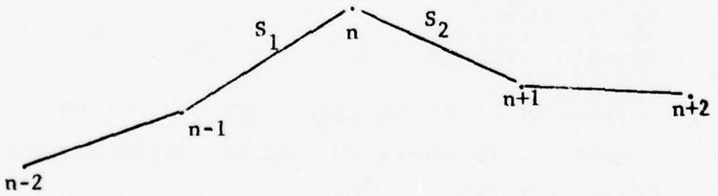
COMMON--/SEARCH/ contains data before and after despiking if NUM = 0

/SMACK/ contains data if NUM = 1

/X/ contains length of the data

/SPIKE/ contains information for PLT1 giving locations where spikes have been removed from each channel

NOTES-- DSPYK finds spikes by considering slopes between successive data points.



If $S_1 * S_2$ is negative, a peak (either high or low) has been passed. If YMIN is the minimum value in the array and YMAX the maximum, then the peak is considered a spike if $S_1 * S_2 < -0.1 * (\text{YMAX-YMIN})^2$. When a spike is detected at n, the tape data are replaced by a straight line connecting n-2 with n+2. The program makes several passes through the data (since YMAX and YMIN may change when a spike is removed) until it makes a pass in which no spikes are found. All spikes found in the HGLP station tapes were only three data points long.

Line 104 + 1 is the statement that determines how sharp a peak must be to be considered a spike.

With the appropriate arguments, PLT1 plots an asterisk at the location of each removed spike.

DSPKY operates on the three channels (CH1-3 or A1-3) in a single CALL statement.

```
      SUBROUTINE DSPYK(NUM)
      COMMON/SEARCH/CH1(5550).CH2(5550).CH3(5550)
      COMMON/X/IXMAX
      COMMON/SMACK/A1(555),A2(555).A3(555).B1(111).B2(111).
     .B3(111).HEAD(10).IREC
      COMMON/SPIKE/SPIK1(500).SPIK2(600).SPIK3(500).ICNT1.ICNT2.
      REAL MASTER
      DIMENSION MASTER(500.3). ICNT(3),IEAD(10)
      DIMENSION YMIN(3).YMAX(3).CH(5550.3).XLIM(3)
      EQUIVALENCE (CH(1.1).CH1(1)).(CH(1.2).CH2(1.)).
     .(CH(1.3).CH3(1))
      EQUIVALENCE(HEAD(1).IEAD(1)).(MASTER(1.1).SPIK1(1))
      EQUIVALENCE(MASTER(1.2).SPIK2(1)).(MASTER(1.3).SPIK3(1))
C
C        INITIALIZE
C
      MAXNUM=500
      DO 75 ICH=1.3
      DO 74 J=1.MAXNUM
 74   MASTER(J.ICH)=0
 75   ICNT(ICH)=0
      ISUM2=0
      IF(NUM.NE.1) GO TO 100
C
C        ROUTINE IF DSPYK IS CALLED FOLLOWING READ
C
      DO 80 J=1.555
      CH1(J)=A1(J)
      CH2(J)=A2(J)
      CH3(J)=A3(J)
 80   CONTINUE
      IXSAVE=IXMAX
      IXMAX=555
C
C        FIND MAX.. MIN.. AND MINIMUM SPIKE SIZE
C
 100  DO 102 ICH=1.3
      YMIN(ICH)=99999.
 102  YMAX(ICH)=-99999.
C
      DO 105 ICH=1.3
      DO 104 J=1.IXMAX
      YMIN(ICH)=AMIN1(YMIN(ICH).CH(J.ICH))
      YMAX(ICH)=AMAX1(YMAX(ICH).CH(J.ICH))
 104  CONTINUE
      XLIM(ICH)=-.10*(YMAX(ICH)-YMIN(ICH))**2
 105  CONTINUE
      DO 800 ICH=1.3
```

```
                     C
                     C          CHECK FOR SPIKES AT BEGINNING OF ARRAY
                     C
ISN 0039                  DO 200 J=2.3
ISN 0040                  SLOPE=(CH(J.ICH)-CH(J-1.ICH))*(CH(J+1.ICH)-CH(J.ICH))
ISN 0041                  IF(SLOPE.GT.XLIM(ICH)) GO TO 200

ISN 0043                  IF(J.EQ.3) CH(5.ICH)=CH(6.ICH)
ISN 0045                  DO 110 L=1.4
ISN 0046              110 CH(L, ICH)=CH(5.ICH)
ISN 0047                  ICNT(ICH)=ICNT(ICH)+1
ISN 0048                  IF(ICNT(ICH).GT.MAXNUM) GO TO 200
ISN 0050                  MASTER(ICNT(ICH).ICH)=J
ISN 0051              200 CONTINUE
                     C
                     C          CHECK FOR SPIKES IN MAIN PART OF ARRAY
                     C
ISN 0052                  JXMAX=IXMAX-3
ISN 0053                  DO 550 J=4.JXMAX
ISN 0054                  J1=J-2
ISN 0055                  J2=J+2
ISN 0056                  SLOPE=(CH(J.ICH)-CH(J-1.ICH))*(CH(J+1.ICH)-CH(J.ICH))
ISN 0057                  IF(SLOPE.GT.XLIM(ICH)) GO TO 550
ISN 0059                  DIFF=CH(J+3.ICH)-CH(J-3.ICH)
ISN 0060                  XCNT=0.
ISN 0061                  DO 300 K=J1.J2
ISN 0062                  XCNT=XCNT+1.
ISN 0063              300 CH(K,ICH)=CH(J-3.ICH)+XCNT*DIFF/6.
ISN 0064                  ICNT(ICH)=ICNT(ICH)+1
ISN 0065                  IF(ICNT(ICH).GT.MAXNUM) GO TO 550
ISN 0067                  MASTER(ICNT(ICH).ICH)=J
ISN 0068              550 CONTINUE
                     C
                     C          CHECK FOR SPIKES IN LAST TWO POINTS IN ARRAY
                     C
ISN 0069                  J1=IXMAX-2
ISN 0070                  J2=IXMAX-1
ISN 0071                  DO 750 J=J1.J2
ISN 0072                  J3=J1-1
ISN 0073                  J4=J1+2
ISN 0074                  SLOPE=(CH(J.ICH)-CH(J-1.ICH))*(CH(J+1.ICH)-CH(J.ICH))
ISN 0075                  IF(SLOPE.GT.XLIM(ICH)) GO TO 750
ISN 0077                  IF(J.EQ.J2) CH(J-2.ICH)=CH(J-3.ICH)
ISN 0079                  DO 590 L=J3.J4
ISN 0080              590 CH(L.ICH)=CH(J-2.ICH)
ISN 0081                  ICNT(ICH)=ICNT(ICH)+1
ISN 0082                  IF(ICNT(ICH).GT.MAXNUM) GO TO 750
ISN 0084                  MASTER(ICNT(ICH .ICH)=J
ISN 0085              750 CONTINUE
ISN 0086              800 CONTINUE
                     C
                     C         CHECK IF NO SPIKES WERE REMOVED IN LAST PASS
                     C
ISN 0087                  ISUM1=ICNT(1)+ICNT(2)+ICNT(3)
ISN 0088                  IF(ISUM1.EQ.ISUM2) GO TO 810
ISN 0090                  ISUM2=ISUM1
ISN 0091                  GO TO 100
ISN 0092              810 IF(NUM.EQ.1) IXMAX=IXSAVE
ISN 0094                  ICNT1=ICNT(1)
ISN 0095                  ICNT2=ICNT(2)
ISN 0096                  ICNT3=ICNT(3)
ISN 0097                  RETURN
ISN 0098                  END
```

NAME--FASTO

TYPE--DATA

SOURCE--G. FRYER

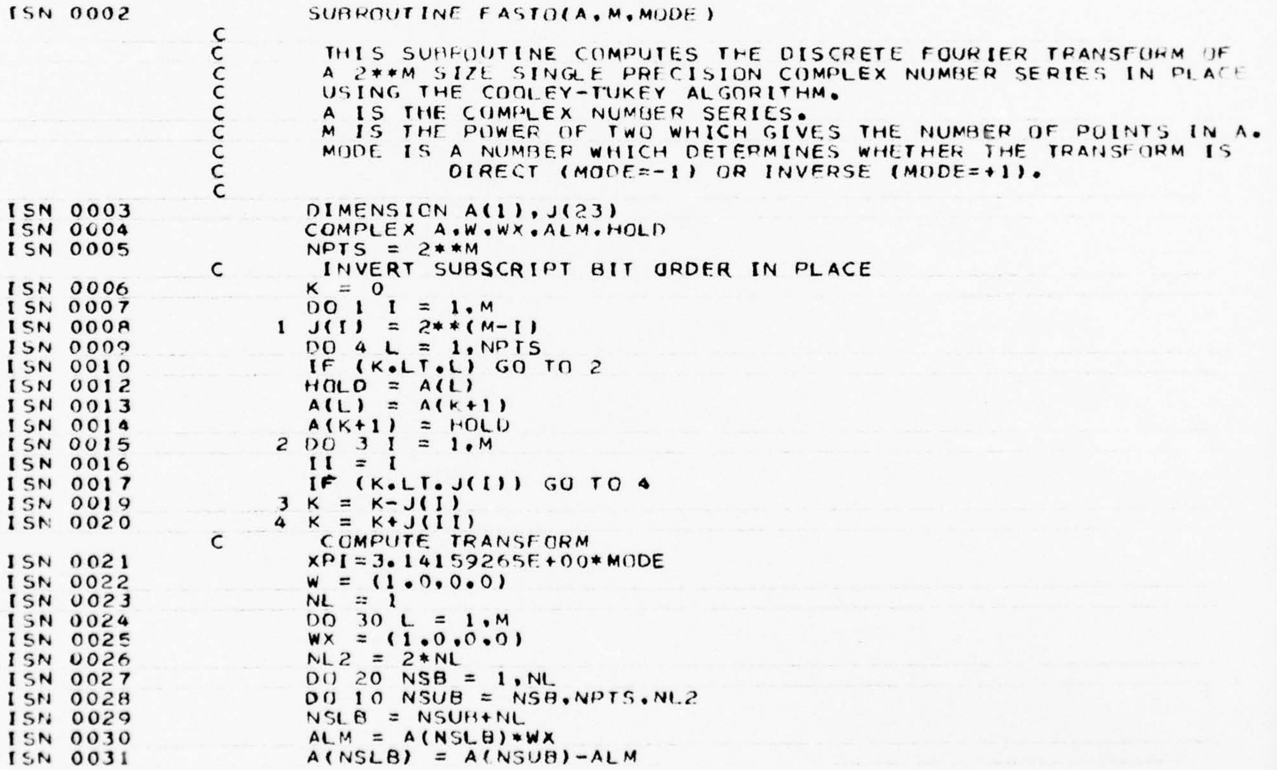
PURPOSE-- Computes the discrete Fourier transform, either direct or inverse, of a complex array of length 2^m where m is an integer.

DESCRIPTION--CALL FASTO (A, M, MODE)

A - complex array

M - power of two, giving length of A

MODE - direct (MODE = -1) or inverse (MODE = +1)



```
ISN 0002            SUBROUTINE FASTO(A,M,MODE)
              C
              C      THIS SUBROUTINE COMPUTES THE DISCRETE FOURIER TRANSFORM OF
              C      A 2**M SIZE SINGLE PRECISION COMPLEX NUMBER SERIES IN PLACE
              C      USING THE COOLEY-TUKEY ALGORITHM.
              C      A IS THE COMPLEX NUMBER SERIES.
              C      M IS THE POWER OF TWO WHICH GIVES THE NUMBER OF POINTS IN A.
              C      MODE IS A NUMBER WHICH DETERMINES WHETHER THE TRANSFORM IS
              C            DIRECT (MODE=-1) OR INVERSE (MODE=+1).
              C
ISN 0003            DIMENSION A(1),J(23)
ISN 0004            COMPLEX A,W,WX,ALM,HOLD
ISN 0005            NPTS = 2**M
              C      INVERT SUBSCRIPT BIT ORDER IN PLACE
ISN 0006            K = 0
ISN 0007            DO 1 I = 1,M
ISN 0008          1 J(I) = 2**(M-I)
ISN 0009            DO 4 L = 1,NPTS
ISN 0010            IF (K.LT.L) GO TO 2
ISN 0012            HOLD = A(L)
ISN 0013            A(L) = A(K+1)
ISN 0014            A(K+1) = HOLD
ISN 0015          2 DO 3 I = 1,M
ISN 0016            II = I
ISN 0017            IF (K.LT.J(I)) GO TO 4
ISN 0019          3 K = K-J(I)
ISN 0020          4 K = K+J(II)
              C      COMPUTE TRANSFORM
ISN 0021            XPI=3.14159265E+00*MODE
ISN 0022            W = (1.0,0.0)
ISN 0023            NL = 1
ISN 0024            DO 30 L = 1,M
ISN 0025            WX = (1.0,0.0)
ISN 0026            NL2 = 2*NL
ISN 0027            DO 20 NSB = 1,NL
ISN 0028            DO 10 NSUB = NSB,NPTS,NL2
ISN 0029            NSLB = NSUB+NL
ISN 0030            ALM = A(NSLB)*WX
ISN 0031            A(NSLB) = A(NSUB)-ALM
ISN 0032          10 A(NSUB) = A(NSUB)+ALM
ISN 0033          20 WX = W*WX
ISN 0034             NL = NL*2
ISN 0035             XPI = XPI/2.0
ISN 0036          30 W = CMPLX(COS(XPI).SIN(XPI))
ISN 0037             IF (MODE.EQ.-1) RETURN
            C         DIVIDE BY NORMALIZATION FACTOR
ISN 0039             DO 50 I = 1,NPTS
ISN 0040          50 A(I) = A(I)/NPTS
ISN 0041             RETURN
            C         END FASTO
ISN 0042             END
```

NAME--FIND

TYPE--IO

SOURCE--D. CHESLEY

PURPOSE-- Reads data tape and returns desired segment of data, A channels only, in COMMON/SEARCH/ with a maximum of 5550 data points.

DESCRIPTION--CALL FIND (ISTART, ISTOP)

ISTART - (dimension = 5, integer) contains year, day, hour, minute, second of first data point desired

ISTOP - (dimension = 5, integer) contains time for last data point

COMMON--/SEARCH/ contains data returned by FIND

CH1 - vertical, CH2-NS, CH3-EW

/SMACK/ - used by READ to supply data to FIND

/X/ - returns length of data in CH1, 2, and 3

NOTES-- FIND locates the desired records on the data tape and calls READ to interpret the tape. FIND then calls SHRINK to eliminate unneeded data. FIND does not despike the data. FIND prints start time of the first record and stop time of the last record, first and last record numbers, and total number of data points (= 555 x numbers of records). These are not the times given by ISTART and ISTOP, but the desired times should be included in the span printed by FIND.

If a record is not readable, FIND replaces missing data with straight lines.

FIND supplies diagnostic messages if an EOF is encountered, or if a record is read more than six times.

```
ISN 0002                SUBROUTINE FIND(ISTART,ISTOP)
ISN 0003                COMMON/SEARCH/CH1(5550),CH2(5550),CH3(5550)
ISN 0004                COMMON/SMACK/A1(555),A2(555),A3(555),B1(111),B2(111),
                       .B3(111),HEAD(10),IREC
ISN 0005                COMMON/X/IXMAX
ISN 0006                INTEGER RECS,REREAD
ISN 0007                DIMENSION IEAD(10),ISTART(1),ISTOP(1),ITIME1(5),ITIME2(5)
ISN 0008                DIMENSION SAVE(10),RECS(3)
ISN 0009                EQUIVALENCE (IEAD(1),HEAD(1))
            C
ISN 0010                DO 10 J=1,3
ISN 0011          10    RECS(J)=0
ISN 0012                IPT=0
ISN 0013                REREAD=0
            C
ISN 0014                CALL HD2SEC(ISTART,ION)
ISN 0015                CALL HD2SEC(ISTOP,IOFF)
ISN 0016                NUMREC=(IOFF-ION)/555+2
            C
ISN 0017                NREC=IREC+1
ISN 0018          20    CALL READ(NREC,IPT)
            C
ISN 0019                IF(IEAD(1).NE.-10) GO TO 30
ISN 0021                NREC=IREC-1
ISN 0022                GO TO 20
ISN 0023          30    RECS(3)=RECS(2)
ISN 0024                RECS(2)=RECS(1)
ISN 0025                RECS(1)=IREC
ISN 0026                IF((RECS(1).NE.RECS(3)).AND.(RECS(1).NE.RECS(2))) GO TO 40
ISN 0028                PRINT 12, RECS(1)
ISN 0029          12    FORMAT(1X,' **** RECORD ',I5,' REREAD ****')
ISN 0030                REREAD=REREAD+1
ISN 0031                IF(REREAD.LT.6) GO TO 40
ISN 0033                PRINT 13
ISN 0034          13    FORMAT(1X,' **** REREAD THRESHOLD REACHED: EXEC TERMINATED
ISN 0035                STOP                                                        ****')
            C
ISN 0036          40    CALL HDCONV(ITIME1,1)
ISN 0037                CALL HD2SEC(ITIME1,IT1)
ISN 0038                ION1=IT1+555
ISN 0039                IF((ION.GE.IT1).AND.(ION.LE.ION1)) GO TO 100
ISN 0041                IS=ION-IT1
ISN 0042                IADD=ISIGN(1,IS)
ISN 0043                IS=IABS(IS)/555
ISN 0044                IF(IS.EQ.0) IS=1
ISN 0046                NREC=IREC+IADD*IS
ISN 0047                IF(NREC.GT.0) GO TO 20
ISN 0049                PRINT 85
ISN 0050          85    FORMAT(1X,'*** WARNING: BEGINNING OF TAPE ***')
ISN 0051                NREC=1
ISN 0052                GO TO 20
            C
ISN 0053         100    IREC1=NREC
ISN 0054                DO 110 J=1,10
ISN 0055                IF((J.EQ.7).OR.(J.EQ.10)) GO TO 108
ISN 0057                SAVE(J)=IEAD(J)
ISN 0058                GO TO 110
ISN 0059         108    SAVE(J)=HEAD(J)
ISN 0060         110    CONTINUE
```

```
                C
ISN 0061                  DO 250 K=1,NUMREC
ISN 0062                  J1=(K-1)*555
ISN 0063                  DO 200 L=1,555
ISN 0064                  J2=J1+L
ISN 0065                  CH1(J2)=A1(L)
ISN 0066                  CH2(J2)=A2(L)
ISN 0067                  CH3(J2)=A3(L)
ISN 0068            200   CONTINUE
ISN 0069                  IF(K.EJ.NUMREC) GO TO 250
ISN 0071                  NREC=NREC+1
ISN 0072                  CALL READ(NREC,IPT)
ISN 0073            250   CONTINUE
                C
ISN 0074                  IREC2=IEAD(1)
ISN 0075                  IXMAX=(IREC2-IREC1+1)*555
ISN 0076                  DO 260 J=1,4
ISN 0077            260   ITIME2(J)=IEAD(J+2)
ISN 0078                  ITIME2(5)=HEAD(7)
ISN 0079                  CALL HD2SEC(ITIME2,IH)
ISN 0080                  IH=IH+555
ISN 0081                  CALL SEC2HD(IH,ITIME2)
                C
ISN 0082                  DO 275 L=1,10
ISN 0083                  IF ((L.EQ.7).OR.(L.EQ.10)) GO TO 274
ISN 0085                  IEAD(L)=SAVE(L)
ISN 0086                  GO TO 275
ISN 0087            274   HEAD(L)=SAVE(L)
ISN 0088            275   CONTINUE
                C
ISN 0089                  DO 350 J=1,IXMAX,555
ISN 0090                  IF(CH1(J).NE.40000) GO TO 350
ISN 0092                  J1=J-1
ISN 0093                  IF (J1.EQ.0) GO TO 310
ISN 0095                  H1=CH1(J1)
ISN 0096                  H2=CH2(J1)
ISN 0097                  H3=CH3(J1)
ISN 0098                  GO TO 330
                C
ISN 0099            310   J1=J+555
ISN 0100                  IF (J1.GT.IXMAX) GO TO 320
ISN 0102                  H1=CH1(J1)
ISN 0103                  H2=CH2(J1)
ISN 0104                  H3=CH3(J1)
ISN 0105                  GO TO 330
                C
ISN 0106            320   H1=0.0
ISN 0107                  H2=0.0
ISN 0108                  H3=0.0
                C
ISN 0109            330   DO 340 J1=1,555
ISN 0110                  J2=J+J1-1
ISN 0111                  CH1(J2)=H1
ISN 0112                  CH2(J2)=H2
ISN 0113                  CH3(J2)=H3
ISN 0114            340   CONTINUE
ISN 0115            350   CONTINUE
                C
ISN 0116                  PRINT 400,IREC1,IREC2,IXMAX,ITIME1,ITIME2
ISN 0117            400   FORMAT(1X,'START RECORD=',I4,', STOP RECORD=',I4,',  ',I7,
                         .' DATA POINTS',/,1X,'START TIME=',I4,'/',I3,'/',I2,':',
                         .I2,':',I2,', STOP TIME=',I4,'/',I3,'/',I2,':',I2,':',I2,/)
ISN 0118                  CALL SHRINK(ISTART,ISTOP)
ISN 0119                  RETURN
ISN 0120                  END
```

NAME--FINDB

TYPE--IO

SOURCE--D. CHESLEY

PURPOSE--Similar to FIND but returns B channels with any desired spacing between successive data points.

DESCRIPTION--CALL FINDB (ISTART, ISTOP, INTER)

ISTART - same as FIND

ISTOP - same as FIND

INTER - desired spacing between data points in seconds. INTER should be an integer multiple of 5.

COMMON--same as FIND

NOTES--FINDB performs the same function as FIND except that FINDB fills CH1, CH2, and CH3 with B-channel (displacement) data. FINDB does not correct for parity and tape reading errors, thus the user must check the data required from the tape prior to using FINDB.

```
ISN 0002              SUBROUTINE FINDB(ISTART,ISTOP,INTER)
ISN 0003              COMMON/SEARCH/CH1(5550),CH2(5550),CH3(5550)
ISN 0004              COMMON/SMACK/A1(555),A2(555),A3(555),B1(111),B2(111),
                     .B3(111),HEAD(10),IREC
ISN 0005              COMMON/X/IXMAX
ISN 0006              INTEGER RECS,REREAD
ISN 0007              DIMENSION IEAD(10),ISTART(1),ISTOP(1),ITIME1(5),ITIME2(5)
ISN 0008              DIMENSION SAVE(10),RECS(3)
ISN 0009              EQUIVALENCE (IEAD(1),HEAD(1))
                C
ISN 0010              DO 10 J=1,3
ISN 0011         10   RECS(J)=0
ISN 0012              REREAD=0
ISN 0013              NEXT=1
                C
ISN 0014              CALL HD2SEC(ISTART,ION)
ISN 0015              CALL HD2SEC(ISTOP,IOFF)
ISN 0016              NUMREC=(IOFF-ION)/555+2
ISN 0017              LEN=(IOFF-ION+1)/INTER+1
                C
ISN 0018              NREC=IREC+1
ISN 0019         20   CALL READ(NREC,0)
                C
ISN 0020              IF(IEAD(1).NE.-10) GO TO 30
ISN 0022              NREC=IREC-1
ISN 0023              GU TO 20
ISN 0024     30    RECS(3)=RECS(2)
ISN 0025           RECS(2)=RECS(1)
ISN 0026           RECS(1)=IREC
ISN 0027           IF((RECS(1).NE.RECS(3)).AND.(RECS(1).NE.RECS(2))) GO TO 40
ISN 0029           PRINT 12, RECS(1)
ISN 0030     12    FORMAT(1X,' **** RECORD ',I5,' REREAD ****')
ISN 0031           REREAD=REREAD+1
ISN 0032           IF(REREAD.LT.6) GO TO 40
ISN 0034           PRINT 13
ISN 0035     13    FORMAT(1X,' **** REREAD THRESHOLD REACHED: EXEC TERMINATED
ISN 0036           STOP                                                    ****')
            C
ISN 0037     40    CALL HDCONV(ITIME1,1)
ISN 0038           CALL HD2SEC(ITIME1,IT1)
ISN 0039           ION1=IT1+555
ISN 0040           IF((ION.GE.IT1).AND.(ION.LE.ION1)) GO TO 100
ISN 0042           IS=ION-IT1
ISN 0043           IADD=ISIGN(1,IS)
ISN 0044           IS=IABS(IS)/555
ISN 0045           IF (IS.EQ.0) IS=1
ISN 0047           NREC=IREC+IADD*IS
ISN 0048           IF(NREC.GT.0) GO TO 20
ISN 0050           PRINT 85
ISN 0051     85    FORMAT(1X,'*** WARNING: BEGINNING OF TAPE ***')
ISN 0052           NREC=1
ISN 0053           GO TO 20
            C
ISN 0054    100    IREC1=NREC
ISN 0055           DO 110 J=1,10
ISN 0056           IF((J.EQ.7).OR.(J.EC.10)) GO TO 108
ISN 0058           SAVE(J)=IEAD(J)
ISN 0059           GO TO 110
ISN 0060    108    SAVE(J)=HEAD(J)
ISN 0061    110    CONTINUE
            C
ISN 0062           ISTEP=INTER/5
ISN 0063           IBEG=(ION-IT1)/INTER+ISTEP
ISN 0064           IEND=111
ISN 0065           DO 250 K=1,NUMREC
ISN 0066           DO 200 J1=IBEG,IEND,ISTEP
ISN 0067           IC=J1
ISN 0068           CH1(NEXT)=B1(J1)
ISN 0069           CH2(NEXT)=B2(J1)
ISN 0070           CH3(NEXT)=B3(J1)
ISN 0071           NEXT=NEXT+1
ISN 0072           IF(NEXT.LE.5550) GO TO 195
ISN 0074           PRINT 150
ISN 0075    150    FORMAT(1X,'*** FIND: DESIRED SCAN EXCEEDS DIMENSION ***',/)
ISN 0076           GO TO 255
ISN 0077    195    IF (NEXT.GT.LEN) GO TO 255
ISN 0079    200    CONTINUE
ISN 0080           IF(K.EQ.NUMREC) GO TO 255
ISN 0082           NREC=IREC+1
ISN 0083           CALL READ(NREC,0)
ISN 0084    201    CONTINUE
ISN 0085           IBEG=ISTEP-IEND+IC
ISN 0086    250    CONTINUE
            C
ISN 0087    255    IREC2=IEAD(1)
ISN 0088           IXMAX=NEXT-1
ISN 0089           IEAD(1)=SAVE(1)
ISN 0090           IEAD(2)=SAVE(2)
ISN 0091           CALL HDCONV(ISTART,-1)
ISN 0092           IEAD(8)=SAVE(8)
ISN 0093           IEAD(9)=SAVE(9)
ISN 0094           HEAD(10)=SAVE(10)
ISN 0095           PRINT 400,IREC1,IREC2,IXMAX,(ISTART(K),K=1,5),(ISTOP(K),K=1,5)
ISN 0096    400    FORMAT(1X,'START RECORD=',I4,', STOP RECORD=',I4,', ',I7,
                  .' DATA POINTS',/,1X,'START TIME=',I4,'/',I3,'/',I2,':',
                  .I2,':',I2,', STOP TIME=',I4,'/',I3,'/',I2,':',I2,':',I2,/)
ISN 0097           RETURN
ISN 0098           END
```

NAME--FLTADJ

TYPE--FILTER

SOURCE--D. CHESLEY

PURPOSE-- Adjusts start and stop times so desired span is still present after the data have been filtered with DOFILT.

DESCRIPTION--CALL FLTADJ (IL, ISRT, ISTP)

IL - length of array containing filter coefficients is 2*IL+1

ISRT - (dimension = 5, integer) contains year, day, hour, minute, second of desired start time after filtering

ISTP - (dimension = 5, integer) contains desired stop time after filtering

FLTADJ changes ISRT and ISTP so that more data than desired will be read from the tape. Subsequent filtering with DOFILT will reduce the data scan to the desired length.

```
      SUBROUTINE FLTADJ(IL,ISRT,ISTP)
      DIMENSION ISRT(1),ISTP(1)
      IN=1
      CALL HD2SEC(ISRT,IS)
      IS=IS-IL*IN
      CALL SEC2HD(IS,ISRT)
      CALL HD2SEC(ISTP,IT)
      IT=IT+IL*IN
      CALL SEC2HD(IT,ISTP)
      RETURN
      END
```

NAME--HAMING

TYPE--FILTER

SOURCE-- D. CHESLEY

PURPOSE-- Places a Hamming window on filter coefficients calculated by HPFILT, LPFILT, or BPFILT.

DESCRIPTION--CALL HAMING (W, N)

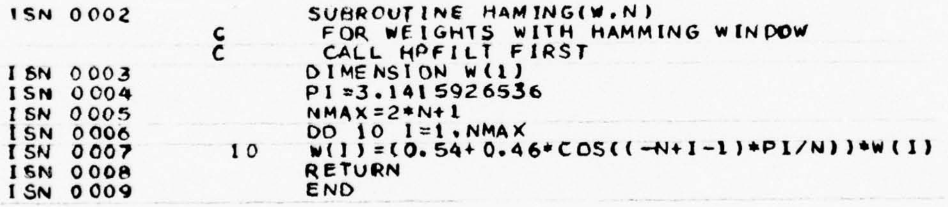
W - the array (real,single precision) containing filter coefficients to be modified by HAMING.

N - the length of W is 2*N+1

The Hamming window has a very sharp cutoff (60 db/octave).

The coefficients are calculated as:

$$W(I) = W(I) \times 0.54 + 0.46 \cos\left[\frac{(-N+I-1)}{N}\right], \; I = 1,2N+1.$$



```
ISN 0002          SUBROUTINE HAMING(W.N)
            C      FOR WEIGHTS WITH HAMMING WINDOW
            C      CALL HPFILT FIRST
ISN 0003          DIMENSION W(1)
ISN 0004          PI=3.1415926536
ISN 0005          NMAX=2*N+1
ISN 0006          DO 10 I=1.NMAX
ISN 0007     10   W(I)=(0.54+0.46*COS((-N+I-1)*PI/N))*W(I)
ISN 0008          RETURN
ISN 0009          END
```

NAME--HDCONV

TYPE--UTILITY

SOURCE--D. CHESLEY

PURPOSE-- converts time array to header array or vice versa.

DESCRIPTION--CALL HDCONV (IHEAD, NUM)

IHEAD- (dimension = 5, integer) contains or will contain year, day, hour, minute, second

NUM = +1 for loading IHEAD

= -1 for loading header array

COMMON--/SMACK/ contains the header array.

NOTES: The header array HEAD(10) is used to carry start times from subroutine to subroutine, but user-supplied times are in arrays such as IHEAD (ISTOP, ISTART in FIND). HDCONV is used to change one type of array into the other.

```
ISN 0002            SUBROUTINE HDCONV(IHEAD,NUM)
ISN 0003            COMMON/SMACK/A1(555),A2(555),A3(555),B1(111),B2(111),
                   .B3(111),HEAD(10),IREC
ISN 0004            DIMENSION IEAD(10),IHEAD(1)
ISN 0005            EQUIVALENCE (IEAD(1),HEAD(1))
ISN 0006            IF (NUM.EQ.-1) GO TO 50
ISN 0008            DO 10 J=1,4
ISN 0009       10   IHEAD(J)=IEAD(J+2)
ISN 0010            IHEAD(5)=HEAD(7)
ISN 0011            GO TO 100
              C
ISN 0012       50   DO 75 J=1,4
ISN 0013       75   IEAD(J+2)=IHEAD(J)
ISN 0014            HEAD(7)=IHEAD(5)
ISN 0015      100   RETURN
ISN 0016            END
```

NAME--HD2SEC

TYPE--UTILITY

SOURCE--D. CHESLEY

PURPOSE--converts start and stop times to seconds since 1968 ignoring leap years (and 1-sec WWV (end of the year) corrections).

DESCRIPTION-- CALL HD2SEC (IA, IB)

IA - (dimension = 5, integer) contains year, day, hour, minute, second, and is not changed by HD2SEC

IB - returned value (integer) in seconds since 1968.

NOTES--SEC2HD performs the opposite function.

```
      SUBROUTINE HD2SEC(IA,IB)
      DIMENSION IA(1)
      IB=(IA(1)-1968)*365*24*3600
      IB=IB+IA(2)*24*3600
      IB=IB+IA(3)*3600+IA(4)*60+IA(5)
      RETURN
      END
```

NAME--HPFILT

TYPE--FILTER

SOURCE--D. CHESLEY

PURPOSE-- Generates coefficients for a high-pass filter to be applied by DOFILT.

DESCRIPTION--CALL HPFILT (W, N, DLT, HPFREQ)

W - an array of 2*N+1 points which contains the calculated coefficients

N - 2*N+1 is the number of coefficients (≤1000)

DLT - digitizing interval of data to be filtered

HPFREQ - the desired high-pass corner frequency (Hz)

COMMON--/FILT/ contains data for PLT1 labels.

```
ISN 0002                  SUBROUTINE HPFILT(W.N.DLT.HPFREQ)
ISN 0003                  COMMON/FILT/FILAB(2),SPER.XLPER
ISN 0004                  DIMENSION XLAB(2)
                C          W IS ARRAY OF 2 N+1 POINTS
                C          N IS NUMBER OF DATA POINTS EACH SIDE OF MIDDLE
                C
ISN 0005                  DIMENSION W(1)
ISN 0006                  DATA XLAB/'HP F','ILT '/
ISN 0007                  FILAB(1)=XLAB(1)
ISN 0008                  FILAB(2)=XLAB(2)
ISN 0009                  SPER=1./HPFREQ
ISN 0010                  PI=3.1415926536
ISN 0011                  CON=DLT*HPFREQ
ISN 0012                  DO 10 I=1,N
ISN 0013                  X=(-N+I-1)*PI*CON*2.
ISN 0014                  W(I)=-2.*CON
ISN 0015                  IF(X.NE.0) W(I)=W(I)*SIN(X)/X
ISN 0017                  W(2*N-I+2)=W(I)
ISN 0018                  W(N+1)=1.-2.*CON
ISN 0019           10     CONTINUE
ISN 0020                  RETURN
ISN 0021                  END
```

NAME-- INFILT

TYPE--FILTER

SOURCE--D. CHESLEY

PURPOSE--Generates coefficients for a filter that (when applied to data by DOFILT) generates original ground motion by deconvolving the instrument response from the seismic data.

DESCRIPTION--CALL INFILT (ICH, W, N, N1, N2)

ICH - channel number of data that will be filtered

1 = Z, 2 = N-S, 3 = E-W

W - array of length 2*N+1 that contains the calculated coefficients

N - 2*N+1 is the number of coefficients

N1 - desired first nonzero Fourier coefficient

N2 - desired last nonzero Fourier coefficeint

NOTES--INFILT reads 256 Fourier coefficients of each of three instrument response curves (Z, NS, EW) from a disk file (no. 14 in JCL). It then sets data points 1 to N1-1 and N2+1 to 256 equal to zero, takes the inverse FFT, and stores the real part of the result in array W backward in positions W(258) to W(513). It sets the remaining values in W to zero. DOFILT then uses this array as a filter in the standard way to retrieve the ground motion. This ground motion may then be filtered, rotated, and so forth, using the subroutines in this report. Refer to the sample program FILTER1 for an example of the use of this subroutine.

```
         SUBROUTINE INFILT(ICH,W,N,N1,N2)
         COMPLEX D1(256),D2(256),D3(256)
         DIMENSION W(1)
         NTWO=8
         READ(14) D1,D2,D3
         REWIND 14
         DO 5 J=1,256
         GO TO (2,3,4),ICH
 2       D1(J)=1./D1(J)
         GO TO 5
 3       D1(J)=1./D2(J)
         GO TO 5
 4       D1(J)=1./D3(J)
 5       CONTINUE
         IF(N1.EQ.1) GO TO 10
         N3=N1-1
         DO 10 J=1,N3
         J1=N-J+1
         D1(J)=CMPLX(0.,0.)
         D1(J1)=CMPLX(0.,0.)
 10      CONTINUE
         N4=N2+1
         N5=N-N2
         DO 20 J=N4,N5
 20      D1(J)=CMPLX(0.,0.)
         CALL FASTO(D1,NTWC,1)
         DO 30 J=1,256
         J1=513-J+1
         W(J1)=REAL(D1(J))
 30      CONTINUE
         DO 40 J=1,257
 40      W(J)=0.
         RETURN
         END
```

NAME--KREAD

TYPE--IO

SOURCE--KARL HINCK

PURPOSE--Reads 7TRK data tapes backward or forward 1 record and returns appropriate error codes in case of read errors. Called only by READ and data must be interpreted by READ.

DESCRIPTION--CALL KREAD (AREA, NBYTES)

AREA - dimensioned greater than the length of a block on the tape

NBYTES - number of bytes read. Set to -40000 if EOF is encountered. Set to -50000 for a read error. An error message is placed in the first 50 bytes of AREA

CALL KRDBK (AREA(LAST), NBYTES)

AREA (LAST) is the end of array AREA

NBYTES - returns 0 if correct

CALL KREAD reads next record on tape

CALL KRDBK backspaces tape one record.

```
KREAD    CSECT
*
* KREAD IS CALLED FROM A FORTRAN PROGRAM AS: CALL KREAD(AREA,NBYTES)...
* AREA MUST BE => BLKSIZE. NBYTES 18 THD NUMBER OF BYTES READ IN....
* FOR END OF FILE NBYTES IS SET TO -40000. FOR A READ ERROR IT IS SETTO
* -50000 AND AN ERROR MESSAGE IS PLACED IN THE FIRST 50 BYTES OF AREA..
* TO BACKSPACE: CALL KRDBK(AREA(LAST),NBYTES). HERE THE 1ST ARG. MUST
* BE THE END OF THE ARRAY AND NBYTES IS 0 IF O.K...
* AFTER THE END OF DATA IS INDICATED, CALLING KREAD AGAIN WILL START AT
* THE BEGINNING AGAIN... ONLY TAPES CAN BE READ BACKWARDS. ANYTHING
* ELSE WILL BE READ FORWARDS.
*
* HIG DEPT OF GEOPHYSICS.....  KARL HINCK....   15MAY74
*
         USING KREAD,15
         MVC   SW1.ADOR
         BAL   15,SAVE
         USING *,15
KRDBK    MVC   SW1,ARDBK
SAVE     SAVE  (14,12)
         DROP  15
         USING KRDBK,4
         LR    4.15
         LA    2,SA
         ST    2,8(13)
         ST    13.SA+4
         LR    13,2
         L     2.SW1
         BR    2
OPEN     LR    3,1
         LA    2.WOPEN
         ST    2.ADOR
         OPEN  INTAPE
         LR    1.3
WOPEN    LM    2,3,0(1)
         READ  CHKT,SF,INTAPE.(2),'S'
         CHECK CHKT
CALC     C     2,=F'-50000'
         BE    BACK
         L     2.INTAPE+68
         LH    2,22(2)
         LNR   2,2
         AH    2,INTAPE+62
BACK     ST    2,0(3)
         L     13,SA+4
         RETURN (14,12),T.RC=0
SA       DS    18F
SW1      DS    F
ADOR     DC    A(DPEN)
ARDBK    DC    A(READBK)
INTAPE   DCB   DSORG=PS,MACRF=(R),EODAD=ENDR,DDNAME=INTAPE,SYNAD=TAPERR
ENDR     L     2,=F'-40000'
         CLOSE INTAPE
         LA    1,OPEN
         ST    1,ADOR
         B     BACK
TAPERR   SYNADAF ACSMETH=BSAM
         ST    14,R14S
         MVC   0(50,2),75(1)
         L     2,=F'-50000'
         SYNADRLS
         L     14,R14S
         BR    14
R14S     DS    F
*
         ENTRY KRDBK
READBK   LM    2,3,0(1)
         READ  CHKTBK,SB,INTAPE.(2),
         CHECK CHKTBK
         B     CALC
         END
```

NAME--LPFILT

TYPE--FILTER

SOURCE--D. CHESLEY

PURPOSE--calculates low-pass filter coefficients for use by DOFILT.

DESCRIPTION--CALL LPFILT (W, N, DLT, BPFREQ)

W - array containing calculated filter coefficients

N - length of W is 2*N+1 (≤1000)

DLT - digitizing interval of array to be filtered

BPFREQ - low pass corner frequency (Hz)

COMMON/FILT/contains data for PLT1 labels.

```
               SUBROUTINE LPFILT(W.N.DLT.BPFREQ)
               COMMON/FILT/FILAB(2).SPER.XLPER
               DIMENSION XLAB(2)
               DIMENSION W(1)
               DATA XLAB/'LP F'.'ILT '/
               FILAB(1)=XLAB(1)
               FILAB(2)=XLAB(2)
               XLPER=1./BPFREQ
               PI=3.14159265 36
               CON=DLT*BPFREQ
               DO 10 I=1.N
               X=(-N+I-1)*PI*CON*2.
               W(I)=2.*CON
               W(I)=W(I)*SIN(X)/X
               W(2*N-I+2)=W(I)
       10      CONTINUE
               W(N+1)=2.*CON
               RETURN
               END
```

NAME--MAXMIN

TYPE--UTILITY

SOURCE--D. CHESLEY

PURPOSE--finds the location and value of maximum and minimum in an array.

DESCRIPTION--CALL MAXMIN (A, N, AMAX, MAXJ, AMIN, MINJ)

A - array to be scanned (real, single precision)

N - length of A

AMAX (AMIN) - returned maximum (minimum) value in A

MAXJ (MINJ) - returned data point number of maximum (minimum)

```
ISN 0002              SUBROUTINE MAXMIN(A,N,AMAX,MAXJ,AMIN,MINJ)
ISN 0003              DIMENSION A(1)
ISN 0004              AMAX=-9999999.
ISN 0005              AMIN=9999999.
ISN 0006              DO 100 J=1,N
ISN 0007              IF(A(J).LE.AMAX) GO TO 50
ISN 0009              MAXJ=J
ISN 0010              AMAX=A(J)
ISN 0011         50   IF(A(J).GE.AMIN) GO TO 100
ISN 0013              MINJ=J
ISN 0014              AMIN=A(J)
ISN 0015        100   CONTINUE
ISN 0016              RETURN
ISN 0017              END
```

NAME--PLOTB

TYPE--IO

SOURCE--D. CHESLEY

PURPOSE--Plots B-channel seismic data with an arbitrary (but multiple of 5) number of seconds between data points.

DESCRIPTION--CALL PLOTB (Z, SCALE, HITE, ZMAX, ZMIN, LKZ, ISPIK, LAB, INTER)

Z through LAB - identical to arguments for PLT1

INTER - spacing between data points in seconds (must be multiple of 5)

NOTES-- Operates exactly the same as PLT1 except that INTER must be specified.

```
      SUBROUTINE PLTB(Z,SCALE,HITE,ZMAX,ZMIN,LKZ,ISPIK,LAB,INTER)
      COMMON/SMACK/A1(555),A2(555),A3(555),B1(111),B2(111),
     .B3(111),HEAD(10),IREC
      COMMON/PLTPAR/XOFF,YOFF,H1,H2
      COMMON/X/IXMAX
      COMMON/SPIKE/SPIK1(500),SPIK2(500),SPIK3(500),ICNT1,ICNT2,ICNT3
      COMMON/ANGLE/ANG,GAIN
      DIMENSION IEAD(10),Z(1)
      DIMENSION X(5550),Y(5550)
      LOGICAL*4 IIHR/':00 '/
      EQUIVALENCE (HEAD(1),IEAD(1))
      KKL=LKZ
      YMAX=ZMAX
      YMIN=ZMIN
      XMAX=IXMAX
      PPM=60./INTER
      XLEN=XMAX*SCALE/(2.54*PPM)
      XXLEN=XLEN+XOFF
      XSCAL=XLEN/XMAX
      IF(ZMAX.NE.7MIN) GO TO 301
C
C       FIND MAXIMUM AND MINIMUM VALUES IN ARRAY
C
      YMAX=-99999.
      YMIN=99999.
      DO 300 J=1,IXMAX
      YMAX=AMAX1(Z(J),YMAX)
      YMIN=AMIN1(Z(J),YMIN)
  300 CONTINUE
      PRINT 299,YMIN,YMAX
  299 FORMAT(1H0,'PLT1: MINIMUM= ',F9.2,3X,'MAXIMUM= ',F9.2//)
C
C      SCALE PLOT AND GENERATE ARRAYS TO BE PLOTTED
C
  301 YYMIN=YMIN
```

```
ISN 0031                  YYMAX=YMAX
ISN 0032                  YMAX=YMAX-YMIN
ISN 0033                  YMIN=0.
ISN 0034                  YSCAL=HITE/YMAX
ISN 0035                  YMEAN=YMAX*YSCAL/2.+YOFF
ISN 0036                  DO 500 J=1,IXMAX
ISN 0037                  X(J)=(J-1)*XSCAL+XOFF
ISN 0038                  Y(J)=YSCAL*(Z(J)-YYMIN)+YOFF
ISN 0039            500   CONTINUE
                  C
                  C          PLOT Y VS. X
                  C
ISN 0040            501   CONTINUE
ISN 0041                  CALL PLOT(X(1),Y(1),3)
ISN 0042                  DO 510 J=2,IXMAX
ISN 0043            510   CALL PLOT(X(J),Y(J),2)
ISN 0044                  CALL PLOT(XXLEN,YMEAN,3)
ISN 0045                  CALL PLCT(XOFF,YMEAN,2)
                  C
                  C          PLOT AND LABEL ONE TICK MARK EVERY 3 CM.
                  C
ISN 0046                  IHR=IEAD(5)
ISN 0047                  REM=HEAD(7)
ISN 0048                  XS=(60.-REM)*XSCAL*PPM/60.+XOFF
ISN 0049                  NMIN=XMAX/PPM
ISN 0050                  NMIN=NMIN+1
ISN 0051                  XINCR=XSCAL*PPM
ISN 0052                  UMEAN=YMEAN+.125
ISN 0053                  BMEAN=UMEAN-0.25
ISN 0054                  YY=BMEAN-.1
ISN 0055                  MIN=IEAD(6)
ISN 0056                  XMINS=0.0
                  C
ISN 0057                  DO 600 J=1,NMIN
ISN 0058                  K=J-1
ISN 0059                  XMIN=XS+XINCR*K
ISN 0060                  IF(MIN.EQ.59) GO TO 520
ISN 0062                  IF((XMIN-XMINS).GE.(3./2.57)) GO TO 520
ISN 0064                  MIN=MIN+1
ISN 0065                  GO TO 600
                  C
ISN 0066            520   XMINS=XMIN
ISN 0067                  IF(XMIN.GT.XXLEN) GO TO 600
ISN 0069                  CALL PLOT(XMIN,UMEAN,3)
ISN 0070                  CALL PLOT(XMIN,BMEAN,2)
ISN 0071                  XX=XMIN-.05
ISN 0072                  MIN=MIN+1
ISN 0073                  IF(MIN.NE.60) GO TO 550
ISN 0075                  IHR=IHR+1
ISN 0076                  IF(IHR.EQ.24) IHR=0
ISN 0078                  XXX=XX-.18
ISN 0079                  XIHR=IHR
ISN 0080                  CALL NUMBFR(XXX,YY,H1,XIHF,0.,-1)
ISN 0081                  CALL SYMBOL(XX,YY,H1,IIHR,0.,3)
ISN 0082                  MIN=0
ISN 0083                  GO TO 600
ISN 0084            550   WMIN=MIN
ISN 0085                  CALL NUMBER(XX,YY,H1,WMIN,0.,-1)
ISN 0086            600   CONTINUE
ISN 0087            601   CONTINUE
                  C
                  C          PLOT *'S AT LOCATIONS WHERE SPIKES WERE REMOVED
                  C
ISN 0088                  IF(ISPIK.EQ.0) GC TC 675
ISN 0090                  IF(ISPIK.EQ.1) ICNT=ICNT1
ISN 0092                  IF(ISPIK.EQ.2) ICNT=ICNT2
ISN 0094                  IF(ISPIK.EO.3) ICNT=ICNT3
ISN 0096                  IF(ICNT.EQ.0) GO TO 675
ISN 0098                  DO 650 I=1,ICNT
ISN 0099                  I1=ICNT-I+1
ISN 0100                  IF(ISPIK.EQ.1) W=(SPIK1(I1)-1)*XSCAL
 SN 0102                  IF(ISPIK.EQ.2) W=(SPIK2(I1)-1)*XSCAL
 SN 0104                  IF(ISPIK.EO.3) W=(SPIK3(I1)-1)*XSCAL
ISN 0106                  W=W+XOFF
ISN 0107                  YHITE=HITE+YOFF
ISN 0108                  CALL SYMBCL(W,YHITE,H1,11,0.,-1)
ISN 0109            650   CONTINUE
ISN 0110            651   CONTINUE
                  C
                  C             PLCT HEADER INFO
                  C
```

```
ISN 0111        675   CONTINUE
ISN 0112              XL=1.5
ISN 0113              YL=0.07
ISN 0114              IF(IEAD(2).EQ.22) GO TO 700
ISN 0116              IF(IEAD(2).EQ.25) GO TO 701
ISN 0118              IF(IEAD(2).EQ.24) GO TO 702
ISN 0120              IF(IEAD(2).EQ.1) GO TO 703
ISN 0122              IF(IEAD(2).EQ.23) GO TO 704
ISN 0124              IF(IEAD(2).EQ.2) GO TO 705
ISN 0126              IF(IEAD(2).EQ.4) GO TO 706
ISN 0128              PRINT 699,IEAD(2)
ISN 0129        699   FORMAT(1X,'STATION I.D. NC. IS: ',I5)
ISN 0130              GO TO 725
ISN 0131        700   CALL SYMBOL(XL,YL,H2,'STATION: (22) KIP.',0.,18)
ISN 0132              GO TO 725
ISN 0133        701   CALL SYMBOL(XL,YL,H2,'STATION: (25) MAT.',0.,18)
ISN 0134              GO TO 725
ISN 0135        702   CALL SYMBOL(XL,YL,H2,'STATION: (24) ZLP.',0.,18)
ISN 0136              GO TO 725
ISN 0137        703   CALL SYMBOL(XL,YL,H2,'STATION: (01) CTA.',0.,18)
ISN 0138              GO TO 725
ISN 0139        704   CALL SYMBOL(XL,YL,H2,'STATICN: (23) ALQ.',0.,18)
ISN 0140              GO TO 725
ISN 0141        705   CALL SYMBOL(XL,YL,H2,'STATION: (02) CHG.',0.,18)
ISN 0142              GO TO 725
ISN 0143        706   CALL SYMBOL(XL,YL,H2,'STATICN: (04) TLO.',0.,18)
ISN 0144        725   XL=XL+19.*6.*H2/7.
ISN 0145        726   CONTINUE
ISN 0146              CALL SYMBOL(XL,YL,H2,'CH:',0.,3)
ISN 0147              XL=XL+3.*6.*H2/7.
ISN 0148              CALL SYMBOL(XL,YL,H2.KKL,0.,2)
ISN 0149              XIN=IEAD(3)
ISN 0150              POS=6.*4.*H2/7.+XL
ISN 0151              CALL NUMBER(POS,YL,H2,XIN,0.,-1)
ISN 0152              CALL WHERE(RETX,RETY,RETFAC)
ISN 0153              POS=RETX
ISN 0154              XIN=-IEAD(4)
ISN 0155              CALL NUMBER(POS,YL,H2,XIN,0.,-1)
ISN 0156              CALL WHERE(RETX,RETY,RETFAC)
ISN 0157              POS=RETX+6.*H2/7.*2.
ISN 0158              XIN=IEAD(5)
ISN 0159              CALL NUMBER(POS,YL,H2,XIN,0.,-1)
ISN 0160              CALL WHERE(RETX,RETY,RETFAC)
ISN 0161              POS=RETX
                C
ISN 0162              IF(IEAD(6).NE.0) GO TO 735
ISN 0164              CALL SYMBOL(POS,YL,H2,IIHR,0.,4)
ISN 0165              POS=POS+4.*6.*H2/7.
ISN 0166              GO TO 738
                C
ISN 0167        735   CALL SYMBOL(POS,YL,H2,122,0.,-1)
ISN 0168              CALL WHERE(RETX,RETY,RETFAC)
ISN 0169              POS=RETX
ISN 0170              XIN=IEAD(6)
ISN 0171              CALL NUMBER(POS,YL,H2,XIN,0.,-1)
ISN 0172              CALL WHERE(RETX,RETY,RETFAC)
ISN 0173              POS=RETX
                C
ISN 0174        738   IF(HEAD(7).NE.0.) GO TO 740
ISN 0176              CALL SYMBOL(POS,YL,H2,IIHR,0.,4)
ISN 0177              GO TO 742
                C
ISN 0178        740   CALL SYMBOL(POS,YL,H2,122,0.,-1)
ISN 0179              CALL WHERE(RETX,RETY,RETFAC)
ISN 0180              POS=RETX
ISN 0181              IIN=HEAD(7)
ISN 0182              XIN=IIN
ISN 0183              CALL NUMBER(POS,YL,H2,XIN,0.,-1)
ISN 0184        742   IF(LAB.EQ.0) GO TO 750
                C
                C       LABEL ROTATION INFO
                C
ISN 0186              POS=POS+.75
ISN 0187              CALL SYMBOL(POS,YL,H2,'ANGLE:',0.,6)
ISN 0188              IPHI=ANG
ISN 0189              XANG=ANG
ISN 0190            POS=POS+6.*6.*H2/7.
ISN 0191            CALL NUMBER(POS,YL,H2,XANG,0.,1)
ISN 0192            CALL WHERE(RETX,RETY,RETFAC)
ISN 0193            POS=RETX
ISN 0194            CALL SYMBOL(POS,YL,H2,'DEG.',0.,4)
             C
ISN 0195            POS=POS+4.*6.*H2/7.
ISN 0196            CALL SYMBOL(POS,YL,H2,'GAIN FACTOR:',0.,12)
ISN 0197            POS=POS+12.*6.*H2/7.
ISN 0198            XGAIN=GAIN
ISN 0199            CALL NUMBER(POS,YL,H2,XGAIN,0.,4)
ISN 0200            CALL WHERE(RETX,RETY,RETFAC)
ISN 0201            POS=RETX
ISN 0202            CALL SYMBOL(POS,YL,H2,'X A3',0.,4)
             C
             C        PLOT AND LABEL Y-AXIS
             C
ISN 0203       750  CONTINUE
ISN 0204       751  CONTINUE
ISN 0205            HYTE=YMIN*YSCAL
ISN 0206            YINCR=YMAX/5.
ISN 0207            NUMDEC=-1
ISN 0208            IF(YINCR.LT.5.) NUMDEC=2
ISN 0210            DO 800 J=1,6
ISN 0211            XK=J-1
ISN 0212            YYY=XK*YINCR*YSCAL+YOFF
ISN 0213            CALL PLOT(0.875,YYY,3)
ISN 0214            CALL PLOT(1.125,YYY,2)
ISN 0215            XYPOS=YYMIN+YINCR*XK
ISN 0216            CALL NUMBER(0.6,YYY,H1,XYPOS,0.,NUMDEC)
ISN 0217       800  CONTINUE
ISN 0218            YYMIN=YMIN+YOFF
ISN 0219            YHITE=HITE+YOFF
ISN 0220            CALL PLOT(1.,YYMIN,3)
ISN 0221            CALL PLOT(1.,YHITE,2)
ISN 0222            CALL PLOT(0.,YHITE,-3)
ISN 0223            CALL PLOT(0.,0.,5)
ISN 0224       801  CONTINUE
ISN 0225            RETURN
ISN 0226            END
```

NAME--PLT1

TYPE--IO

SOURCE--D. CHESLEY

PURPOSE--Plots seismic data on a XYNETICS plotter. Labels each plot with station name and code, start time, rotation, and filter information. Data points must be one per second.

DESCRIPTION--CALL PLT1 (Z, SCALE, HITE, ZMAX, ZMIN, LKZ, ISPIK, LAB)

Z - the array to be plotted

SCALE - horizontal scale in cm/min

HITE - height of the plot (without labels) in inches

ZMAX - maximum value of Y-axis

ZMIN - minimum value of Y-axis

LKZ - two-character string enclosed in apostrophes, used to label the channel that is plotted ('A1', 'A2', 'SM')

ISPIK - channel number (1, 2, or 3 for A1, A2, A3 respectively) causes asterisks to be plotted at each location where DSPYK has removed a spike. ISPIK = 0 means no asterisks are plotted.

LAB - label parameter:

0 - no rotation and no filter

1 - rotation but no filter

2 - filter but no rotation

3 - both filter and rotation information are plotted in the label

COMMON--/SMACK/ contains HEAD(10), which holds the start time of the trace

/PLTPAR/ holds letter sizes and X and Y offsets (see SET)

/FILT/ contains filter information for label

/X/ contains length of Z

/SPIKE/ contains positions of spikes which were removed by DSPYK

/ANGLE/ contains rotation information for label

NOTES-- PLT1 is designed specifically for plotting A-channel data at 1 data point per second. HEAD(1-10) contains the start time. PLT1 plots and labels minute marks, except that PLT1 suppresses the marks if they are less than three centimeters from the previous mark, to avoid overcrowding. Hour marks are always plotted. PLT1 interprets the station number, HEAD(2), if possible and labels the plot with a three-letter abbreviation. If the station number is not decodable it is printed and that portion of the plot label is deleted. With the appropriate value of LAB, the subroutine includes rotation and/or filter information in the label. The Y-axis is labelled automatically. If the year (IEAD(3)) is zero the time is not plotted. The desired maximum and minimum values to be plotted may be defined in the argument list (ZMAX, ZMIN), and thus the vertical scale is user-determined. However, if ZMAX = ZMIN, PLT1 scans the array and uses the maximum and minimum values in the array to define the vertical scale. In this case the maximum and minimum are printed.

The starting position of the pen is (-XOFF, -OFF-H2) from the lower end of the Y-axis. The final position is (-XOFF, HITE+H2). Changes in SCALE and HITE will not affect the letter size.

PLT1 requires the statement CALL SETXYN (or SETCC) prior to the plotting command and the statement

CALL PLOT (0, HITE, 999)

at the end of the main program. An output tape must be supplied. See the sample programs for JCL details.

CALL PLT1 generates ones XYNETICS drawing.

```
ISN 0002              SUBROUTINE PLT1 (Z,SCALE,HITE,ZMAX,ZMIN,LKZ,ISPIK,LAB)
ISN 0003              COMMON/SMACK/A1(555),A2(555),A3(555),B1(111),B2(111),
                      B3(111),HEAD(10),IREC
ISN 0004              COMMON/PLTPAR/XOFF,YOFF,H1,H2
ISN 0005              COMMON/FILT/FILAB(2),SPER,XLPER
ISN 0006              COMMON/X/IXMAX
ISN 0007              COMMON/SPIKE/SPIK1(500),SPIK2(500),SPIK3(500),ICNT1,ICNT2,ICNT3
ISN 0008              COMMON/ANGLE/ANG,GAIN
ISN 0009              DIMENSION IEAD(10),Z(1)
ISN 0010              DIMENSION X(5550),Y(5550)
ISN 0011              LOGICAL*4 FILAB
ISN 0012              LOGICAL*4 IIHR/':00 '/
ISN 0013              EQUIVALENCE (HEAD(1),IEAD(1))
ISN 0014              WTH=6.*H2/7.
ISN 0015              ZERO=0.
ISN 0016              KKL=LKZ
ISN 0017              YMAX=ZMAX
ISN 0018              YMIN=ZMIN
ISN 0019              XMAX=IXMAX
ISN 0020              XLEN=XMAX*SCALE/(2.54*60.)
ISN 0021              XXLEN=XLEN+XOFF
ISN 0022              XSCAL=XLEN/XMAX
ISN 0023              IF(ZMAX.NE.ZMIN) GO TO 301
                C
                C       FIND MAXIMUM AND MINIMUM VALUES IN ARRAY
                C
ISN 0025              YMAX=-99999.
ISN 0026              YMIN=99999.
ISN 0027              DO 300 J=1,IXMAX
ISN 0028              YMAX=AMAX1(Z(J),YMAX)
ISN 0029              YMIN=AMIN1(Z(J),YMIN)
ISN 0030          300 CONTINUE
ISN 0031              PRINT 299,YMIN,YMAX
ISN 0032          299 FORMAT(1H0,'PLT1: MINIMUM= ',E9.3,3X,'MAXIMUM= ',E9.3//)
                C
                C      SCALE PLOT AND GENERATE ARRAYS TO BE PLOTTED
                C
ISN 0033          301 YYMIN=YMIN
ISN 0034              YYMAX=YMAX
ISN 0035              YMAX=YMAX-YMIN
ISN 0036              YMIN=0.
ISN 0037              YSCAL=HITE/YMAX
ISN 0038              YMEAN=YMAX*YSCAL/2.+YOFF
ISN 0039              DO 500 J=1,IXMAX
ISN 0040              X(J)=(J-1)*XSCAL+XOFF
ISN 0041              Y(J)=YSCAL*(Z(J)-YYMIN)+YOFF
ISN 0042          500 CONTINUE
```

```
                    C
                    C           PLOT Y VS. X
                    C
ISN 0043              501   CONTINUE
ISN 0044                    CALL PLOT(X(1),Y(1),3)
ISN 0045                    DO 510 J=2,IXMAX
ISN 0046              510   CALL PLOT(X(J),Y(J),2)
ISN 0047                    CALL PLOT(XXLEN,YMEAN,3)
ISN 0048                    CALL PLOT(XOFF,YMEAN,2)
                    C
                    C           PLOT AND LABEL ONE TICK MARK EVERY 3 CM.
                    C
ISN 0049                    IHR=IEAD(5)
ISN 0050                    REM=HEAD(7)
ISN 0051                    XS=(60.-REM)*XSCAL+XOFF
ISN 0052                    NMIN=XMAX/50.
ISN 0053                    NMIN=NMIN+1
ISN 0054                    XINCR=XSCAL*60.
ISN 0055                    UMEAN=YMEAN+.125
ISN 0056                    BMEAN=UMEAN-0.25
ISN 0057                    YY=BMEAN-.1
ISN 0058                    MIN=IEAD(6)
ISN 0059                    XMINS=0.0
                    C
ISN 0060                    DO 600 J=1,NMIN
ISN 0061                    K=J-1
ISN 0062                    XMIN=XS+XINCR*K
ISN 0063                    IF(MIN.EQ.59) GO TO 520
ISN 0065                    IF((XMIN-XMINS).GE.(3./2.57)) GO TO 520
ISN 0067                    MIN=MIN+1
ISN 0068                    GO TO 600
                    C
ISN 0069              520   XMINS=XMIN
ISN 0070                    IF(XMIN.GT.XXLEN) GO TO 600
ISN 0072                    CALL PLOT(XMIN,UMEAN,3)
ISN 0073                    CALL PLOT(XMIN,BMEAN,2)
ISN 0074                    XX=XMIN-.05
ISN 0075                    MIN=MIN+1
ISN 0076                    IF(MIN.NE.60) GO TO 550
ISN 0078                    IHR=IHR+1
ISN 0079                    IF(IHR.EQ.24) IHR=0
ISN 0081                    XXX=XX-.18
ISN 0082                    XIHR=IHR
ISN 0083                    CALL NUMBER(XXX,YY,H1,XIHR,0.,-1)
ISN 0084                    CALL SYMBOL(XX,YY,H1,IIHR,0.,3)
ISN 0085                    MIN=0
ISN 0086                    GO TO 600
ISN 0087              550   WMIN=MIN
ISN 0088                    CALL NUMBER(XX,YY,H1,WMIN,0.,-1)
ISN 0089              600   CONTINUE
ISN 0090              601   CONTINUE
                    C
                    C           PLOT *'S AT LOCATIONS WHERE SPIKES WERE REMOVED
                    C
ISN 0091                    IF(ISPIK.EQ.0) GO TO 675
ISN 0093                    IF(ISPIK.EQ.1) ICNT=ICNT1
ISN 0095                    IF(ISPIK.EQ.2) ICNT=ICNT2
ISN 0097                    IF(ISPIK.EQ.3) ICNT=ICNT3
ISN 0099                    IF(ICNT.EQ.0) GO TO 675
ISN 0101                    DO 650 I=1,ICNT
ISN 0102                    I1=ICNT-I+1
ISN 0103                    IF(ISPIK.EQ.1) W=(SPIK1(I1)-1)*XSCAL
ISN 0105                    IF(ISPIK.EQ.2) W=(SPIK2(I1)-1)*XSCAL
ISN 0107                    IF(ISPIK.EQ.3) W=(SPIK3(I1)-1)*XSCAL
ISN 0109                    W=W+XOFF
ISN 0110                    YHITE=HITE+YOFF
ISN 0111                    CALL SYMBOL(W,YHITE,H1,11,0.,-1)
ISN 0112              650   CONTINUE
ISN 0113              651   CONTINUE
                    C
                    C             PLOT HEADER INFO
                    C
ISN 0114              675   CONTINUE
ISN 0115             IF(IEAD(2).EQ.0) GO TO 725
ISN 0117             XL=1.5
ISN 0118             YL=0.07
ISN 0119             IF(IEAD(2).EQ.22) GO TO 700
ISN 0121             IF(IEAD(2).EQ.25) GO TO 701
ISN 0123             IF(IEAD(2).EQ.24) GO TO 702
ISN 0125             IF(IEAD(2).EQ.1) GO TO 703
ISN 0127             IF(IEAD(2).EQ.23) GO TO 704
ISN 0129             IF(IEAD(2).EQ.2) GO TO 705
ISN 0131             IF(IEAD(2).EQ.4) GO TO 706
ISN 0133             PRINT 699,IEAD(2)
ISN 0134        699  FORMAT(1X,'STATION I.D. NO. IS: ',I5)
ISN 0135             GO TO 725
ISN 0136        700  CALL SYMBOL(XL,YL,H2,'ST: (22) KIP.',0.,13)
ISN 0137             GO TO 725
ISN 0138        701  CALL SYMBOL(XL,YL,H2,'ST: (25) MAT.',0.,13)
ISN 0139             GO TO 725
ISN 0140        702  CALL SYMBOL(XL,YL,H2,'ST: (24) ZLP.',0.,13)
ISN 0141             GO TO 725
ISN 0142        703  CALL SYMBOL(XL,YL,H2,'ST: (01) CTA.',0.,13)
ISN 0143             GO TO 725
ISN 0144        704  CALL SYMBOL(XL,YL,H2,'ST: (23) ALQ.',0.,13)
ISN 0145             GO TO 725
ISN 0146        705  CALL SYMBOL(XL,YL,H2,'ST: (02) CHG.',0.,13)
ISN 0147             GO TO 725
ISN 0148        706  CALL SYMBOL(XL,YL,H2,'ST: (04) TLD.',0.,13)
ISN 0149        725  XL=XL+14.*WTH
ISN 0150             CALL SYMBOL(XL,YL,H2,'CH:',0.,3)
ISN 0151             XL=XL+3.*WTH
ISN 0152             CALL SYMBOL(XL,YL,H2,KKL,0.,2)
ISN 0153             IF(IEAD(3).EQ.0) GO TO 742
ISN 0155             XIN=IEAD(3)
ISN 0156             POS=XL+4.*WTH
ISN 0157             CALL NUMBER(POS,YL,H2,XIN,0.,-1)
ISN 0158             CALL WHERE(RETX,RETY,RETFAC)
ISN 0159             POS=RETX
ISN 0160             XIN=-IEAD(4)
ISN 0161             CALL NUNBER(POS,YL,H2,XIN,0.,-1)
ISN 0162             CALL WHERE(RETX,RETY,RETFAC)
ISN 0163             POS=RETX+2.*WTH
ISN 0164             XIN=IEAD(5)
ISN 0165             CALL NUMBER(POS,YL,H2,XIN,0.,-1)
ISN 0166             CALL WHERE(RETX,RETY,RETFAC)
ISN 0167             POS=RETX
               C
ISN 0168             IF(IEAD(6).NE.0) GO TO 735
ISN 0170             CALL SYMBOL(POS,YL,H2,IIHR,0.,4)
ISN 0171             POS=POS+3.*WTH
ISN 0172             GO TO 738
               C
ISN 0173        735  CALL SYMBOL(POS,YL,H2,122,0.,-1)
ISN 0174             CALL WHERE(RETX,RETY,RETFAC)
ISN 0175             POS=RETX
ISN 0176             XIN=IEAD(6)
ISN 0177             IF(XIN.GT.9.) GO TO 736
ISN 0179             CALL NUMBER(POS,YL,H2,ZERC,0.,-1)
ISN 0180             CALL WHERE(RETX,RETY,RETFAC)
ISN 0181             POS=RETX
ISN 0182        736  CALL NUMBER(POS,YL,H2,XIN,0.,-1)
ISN 0183             CALL WHERE(RETX,RETY,RETFAC)
ISN 0184             POS=RETX
               C
ISN 0185        738  IF(HEAD(7).NE.0.) GO TO 740
ISN 0187             CALL SYMBOL(POS,YL,H2,IIHR,0.,4)
ISN 0188             GO TO 742
               C
ISN 0189        740  CALL SYMBOL(POS,YL,H2,122,0.,-1)
ISN 0190             CALL WHERE(RETX,RETY,RETFAC)
ISN 0191             POS=RETX
ISN 0192             IIN=HEAD(7)
ISN 0193             IF(IIN.GT.9) GO TO 741
ISN 0195             CALL NUMBER(POS,YL,H2,ZERO,0.,-1)
ISN 0196             CALL WHERE(RETX,RETY,RETFAC)
ISN 0197             POS=RETX
ISN 0198        741  XIN=IIN
ISN 0199             CALL NUMBER(POS,YL,H2,XIN,0.,-1)
ISN 0200        742  ILAB=LAB+1
ISN 0201             GO TO(750,744,745,744),ILAB
```

```
                C
                C          LABEL ROTATION INFO
                C
ISN 0202         744  CALL WHERE(RETX.RETY.RETFAC)
ISN 0203              POS=RETX+2.*WTH
ISN 0204              CALL SYMBOL(POS.YL.H2.'ANGLE:'.0..6)
ISN 0205              XANG=ANG
ISN 0206              POS=POS+6.*WTH
ISN 0207              CALL NUMBER(POS.YL.H2.XANG.0..1)
ISN 0208              CALL WHERE(RETX.RETY.RETFAC)
ISN 0209              POS=RETX+WTH
ISN 0210              CALL SYMBOL(POS.YL.H2.'DEG.'.0..4)
                C
ISN 0211              POS=POS+5.*WTH
ISN 0212              CALL SYMBOL(POS.YL.H2.'GAIN FACTOR:'.0..12)
ISN 0213              POS=POS+12.*WTH
ISN 0214              XGAIN=GAIN
ISN 0215              CALL NUMBER(POS.YL.H2.XGAIN.0..4)
ISN 0216              CALL WHERE(RETX.RETY.RETFAC)
ISN 0217              POS=RETX
ISN 0218              CALL SYMBOL(POS,YL,H2,'X A3'.0..4)
ISN 0219              IF(ILAB.EQ.2) GO TO 750
                C
                C            LABEL FILTER INFO
                C
ISN 0221         745  CALL WHERE(RETX.RETY.RETFAC)
ISN 0222              POS=RETX+2.*WTH
ISN 0223              CALL SYMBOL(POS.YL.H2.FILAB.0..8)
ISN 0224              CALL WHERE(RETX.RETY.RETFAC)
ISN 0225              Y2=RETY
ISN 0226              POS=RETX
ISN 0227              IF(SPER.EQ.9999.) GO TO 746
ISN 0229              CALL NUMBER(POS.Y2.H2.SPER.0..-1)
ISN 0230              CALL WHERE(RETX.RETY.RETFAC)
ISN 0231              POS=RETX
ISN 0232              IF(XLPER.EQ.9999.) GO TO 747
ISN 0234         746  XXLPER=-XLPER
ISN 0235              IF(SPER.EQ.9999.) XXLPER=-XXLPER
ISN 0237              CALL NUMBER(POS.Y2.H2.XXLPER.0..-1)
ISN 0238              CALL WHERE(RETX.RETY.RETFAC)
ISN 0239              POS=RETX
ISN 0240         747  CALL SYMBOL(POS.Y2.H2.'SEC'.0..3)
                C
                C         PLOT AND LABEL Y-AXIS
                C
ISN 0241         750  CONTINUE
ISN 0242              HYTE=YMIN*YSCAL
ISN 0243              YINCR=YMAX/5.
ISN 0244              NUMDEC=-1
ISN 0245              IF(YINCR.LT.5.) NUMDEC=2
ISN 0247              DO 800 J=1.6
ISN 0248              XK=J-1
ISN 0249              YYY=XK*YINCR*YSCAL+YOFF
ISN 0250              CALL PLOT(0.875.YYY.3)
ISN 0251              CALL PLOT(1.125.YYY.2)
ISN 0252              XYPOS=YYMIN+YINCR*XK
ISN 0253              CALL NUMBER(0.5.YYY.H1.XYPOS.0..NUMDEC)
ISN 0254         800  CONTINUE
ISN 0255              YYMIN=YMIN+YOFF
ISN 0256              YHITE=HITE+YOFF
ISN 0257              CALL PLOT(1..YYMIN.3)
ISN 0258              CALL PLOT(1..YHITE.2)
ISN 0259              CALL PLOT(0..YHITE.-3)
ISN 0260              CALL PLOT(0..0..5)
ISN 0261         801  CONTINUE
ISN 0262              RETURN
ISN 0263              END
```

NAME--PRPLT1

TYPE--IO

SOURCE--A. LAZAREWICZ

PURPOSE -- Generates a plot of an array on the high-speed line printer.

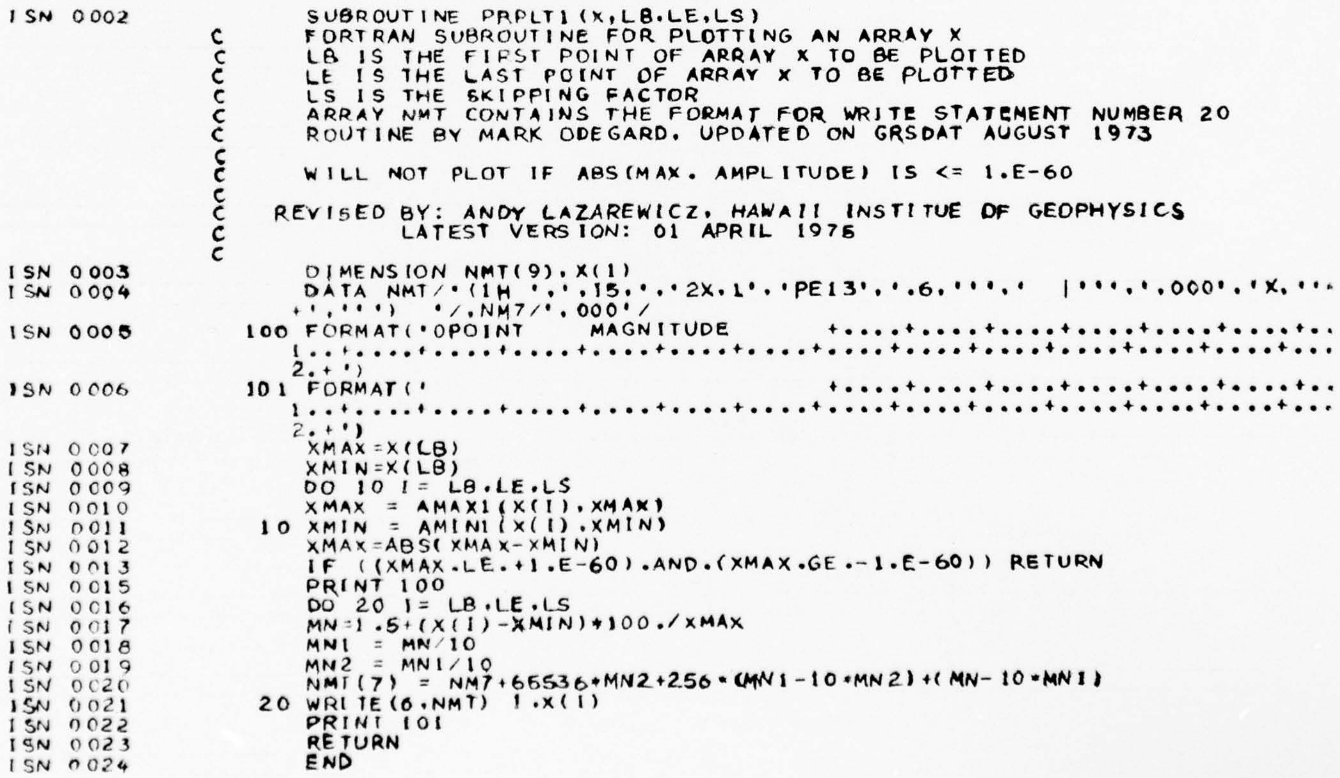
DESCRIPTION--CALL PRPLT1 (X, LB, LE, LS)

X - array to be plotted (single precision, real)

LB - first point to be plotted

LE - last point to be plotted

LS - point increment (1 means plot every point, 2 means plot every other point, and so on)



```
ISN 0002              SUBROUTINE PRPLT1(X,LB.LE,LS)
               C      FORTRAN SUBROUTINE FOR PLOTTING AN ARRAY X
               C      LB IS THE FIRST POINT OF ARRAY X TO BE PLOTTED
               C      LE IS THE LAST POINT OF ARRAY X TO BE PLOTTED
               C      LS IS THE SKIPPING FACTOR
               C      ARRAY NMT CONTAINS THE FORMAT FOR WRITE STATEMENT NUMBER 20
               C      ROUTINE BY MARK ODEGARD. UPDATED ON GRSDAT AUGUST 1973
               C
               C      WILL NOT PLOT IF ABS(MAX. AMPLITUDE) IS <= 1.E-60
               C
               C    REVISED BY: ANDY LAZAREWICZ. HAWAII INSTITUE OF GEOPHYSICS
               C           LATEST VERSION: 01 APRIL 1976
               C
ISN 0003              DIMENSION NMT(9).X(1)
ISN 0004              DATA NMT/'(1H ','',I5.','2X.1','PE13','.6,''','  |''','.000','X,''',
                     +',''')  '/,NM7/'.000'/
ISN 0005          100 FORMAT('0POINT    MAGNITUDE        +....+....+....+....+....+....+..
                     1..+....+....+....+....+....+....+....+....+....+....+....+...
                     2.+')
ISN 0006          101 FORMAT('                            +....+....+....+....+....+....+..
                     1..+....+....+....+....+....+....+....+....+....+....+....+...
                     2.+')
ISN 0007              XMAX=X(LB)
ISN 0008              XMIN=X(LB)
ISN 0009              DO 10 I= LB.LE.LS
ISN 0010              XMAX = AMAX1(X(I).XMAX)
ISN 0011           10 XMIN = AMIN1(X(I).XMIN)
ISN 0012              XMAX=ABS(XMAX-XMIN)
ISN 0013              IF ((XMAX.LE.+1.E-60).AND.(XMAX.GE.-1.E-60)) RETURN
ISN 0015              PRINT 100
ISN 0016              DO 20 I= LB.LE.LS
ISN 0017              MN=1.5+(X(I)-XMIN)*100./XMAX
ISN 0018              MN1 = MN/10
ISN 0019              MN2 = MN1/10
ISN 0020              NMT(7) = NM7+65536*MN2+256*(MN1-10*MN2)+(MN-10*MN1)
ISN 0021           20 WRITE(6.NMT) I.X(I)
ISN 0022              PRINT 101
ISN 0023              RETURN
ISN 0024              END
```

NAME--READ

TYPE--IO

SOURCE--A. LAZAREWICZ

PURPOSE--Reads and interprets one record from 7-TRK data tape.
Fills six arrays with data and supplies header information.
Interprets error codes from KREAD.

DESCRIPTION--CALL READ (NREC, IPRT)

NREC - the number of the record to be read

IPRT - print code

0 - print nothing (except errors), fill arrays

1 - print one-line header, fill arrays

2 - print long header, fill arrays, print arrays

3 - print one-line header, do not fill arrays

4 - print nothing (except errors), do not fill arrays

COMMON/SMACK/ contains data from tape:

A1 - vertical velocity channel

A2 - NS vel. channel

A3 - EW vel. channel

B1 - vertical displacement channel

B2 - NS disp. channel

B3 - EW disp. channel

HEAD(10) - header information (see listing)

IREC - number of the last record that was read

NOTES--In general, the only time a main program calls READ directly is to skip part of a data tape where there is a long (several days) section of data missing. For example, consider a tape that contains the following sets of data in one file:

1975-323-00:00:00 to 1975-326-15:00:00 and

1975-340-12:00:00 to 1975-350-03:00:00

A call to FIND with ISTART as a time in the second set of data may result in an error. FIND reads the first record and uses that start time to calculate the record number that corresponds to ISTART. Because of the gap, the calculated record number could be larger than the number of records in the file and an error may result. The solution is to CALL READ (NREC, 4), where NREC is a record number in the second stretch of data but before the start of the data to be read from the tape. This merely advances the tape past the gap. A subsequent CALL FIND will work properly.

```
ISN 0002             SUBROUTINE READ(NREC,IPRT)
               C
               C        THIS IS A TAPE READING PROGRAM FOR THE RAYLEIGH WAVES PROJECT.
               C        SUMMARY OF TAPE: 7 TRACK, 6006 6-BIT BYTES, ONE FILE PER TAPE.
               C        PLEASE SEE DOCUMENTATION ON USE OF THE PROGRAM, AS SPECIFIC
               C        INTERACTION WITH THE CALLING PROGRAM IS REQUIRED.  LATEST
               C        VERSION: 16 MAY 1974.
               C        WRITTEN BY ANDY LAZAREWICZ, HIG-215 X-8143.
               C
ISN 0003             INTEGER*2 CH,IM(3500),IN(111),ICNV(6),IPAR(1998),LAP(2),ERR(25)
ISN 0004             DIMENSION IEAD(10),IBUF(1998),IA1(555),IA2(555),IA3(555),
                    .IB1(111),IB2(111),IB3(111)
ISN 0005             COMMON/SMACK/A1(555),A2(555),A3(555),B1(111),B2(111),B3(111),
                    .HEAD(10),IREC
ISN 0006             EQUIVALENCE(HEAD(1),IEAD(1)),(KAP,LAP(1)),(A1(1),IA1(1)),
                    .(A2(1),IA2(1)),(A3(1),IA3(1)),(B1(1),IB1(1)),(B2(1),IB2(1)),
                    .(B3(1),IB3(1)),(IM(1),ERR(1))
ISN 0007          1   FORMAT(15(200A2),3A2)
ISN 0008          2   FORMAT(' *** DATA PARITY ERROR AT ',Z2,' - ',I3,' ***')
ISN 0009          3   FORMAT(' NORMALIZED LISTING IN VOLTS FOR RECORD ',I4,/,' STATION '
                    .,I2,', DATE: ',I4,'-',I3,', TIME: ',I2,':',I2,':',F4.1,/,' ',I1,'
                    .A CHANNELS, ',I1,' B CHANNELS, SAMPLE RATE = ',F3.1,//)
ISN 0010          4   FORMAT(I4,1X,15(2X,I6))
ISN 0011          5   FORMAT(' RECORD ',I4,' HAS ',I4,' BYTES, SHOULD BE 6006. DATA SKIP
                    .PED.')
ISN 0012          6   FORMAT(I4,1X,3(2X,I6))
ISN 0013          7   FORMAT(' SEC',2X,5(4X,'A1',6X,'A2',6X,'A3',2X))
ISN 0014         8    FORMAT(' SEC',2X,5(4X,'B1',6X,'B2',6X,'B3',2X))
ISN 0015         9    FORMAT(' RECORD ',I4,', STATION ',I2,', DATE & TIME: ',I4,'-',I3,'
                     .-',I2,':',I2,':',F4.1,', (',I1,',',I1,'), (A,B) CHANNELS, SMAPLE R
                     .ATE: ',F3.1)
ISN 0016        10    FORMAT(' END OF FILE, ',I4,' RECORDS, DESIRED RECORD WAS ',I4)
ISN 0017        12    FORMAT(' ***** READ ERROR IN RECORD ',I4,': ',25A2,', RECORD SKIPP
                     .ED *****')
                C
                C        CHOOSE AND READ DESIRED RECORD
                C
ISN 0018       104    LREC=NREC-IREC
ISN 0019              IF (LREC.GT.0) GO TO 103
ISN 0021              CALL KRDBK(IM(3500),IBYTE)
ISN 0022              IREC=IREC-1
ISN 0023              GO TO 104
ISN 0024       103    DO 11 J=1,LREC
ISN 0025              IREC=IREC+1
ISN 0026              CALL KREAD(IM,IBYTE)
ISN 0027              IF (IBYTE.NE.-40000) GO TO 101
ISN 0029              MREC=IREC-1
ISN 0030              WRITE(6,10) MREC,NREC
ISN 0031              IEAD(1)=-10
ISN 0032              GO TO 200
ISN 0033       101    IF (IBYTE.NE.-50000) GO TO 102
ISN 0035              WRITE(6,12) IREC,ERR
ISN 0036              IF(IREC.EQ.NREC) GO TO 200
ISN 0038              GO TO 11
ISN 0039       102    IF (IBYTE.EQ.6006) GO TO 11
ISN 0041              WRITE (6,5) IREC,IBYTE
ISN 0042              IF (IPRT.EQ.2) IPRT=1
ISN 0044        11    CONTINUE
ISN 0045              DO 15 J=1,6
ISN 0046              K1=IM(J)/256
ISN 0047              K2=IM(J)-K1*256
ISN 0048              DO 16 J1=1,6
ISN 0049              L1=K2/2
ISN 0050              L2=L1*2
ISN 0051              J2=(J-1)*12+13-J1
ISN 0052              IN(J2)=K2-L2
ISN 0053        16    K2=L1
ISN 0054              DO 17 J1=1,6
ISN 0055              L1=K1/2
ISN 0056              L2=L1*2
ISN 0057              J2=(J-1)*12+7-J1
ISN 0058              IN(J2)=K1-L2
ISN 0059        17    K1=L1
ISN 0060        15    CONTINUE
                C
                C        INTERPRET HEADER AND FORM FIXED/FLOAT HEAD ARRAY
                C        1:RELATIVE RECORD, 2:STATION ID, 3:YEAR, 4:JULIAN DAY, 5:HOUR,
                C        6:MINUTES, 7:SECONDS, 8:NUMBER OF A CHANNELS, 9:NUMBER OF B
                C        CHANNELS, 10:SAMPLE RATE.
                C
ISN 0061              IEAD(1)=IREC
ISN 0062              IEAD(2)=10*(8*IN(3)+4*IN(4)+2*IN(5)+IN(6))+8*IN(9)+4*IN(10)+
                     .2*IN(11)+IN(12)
ISN 0063              IEAD(3)=1970.+8*IN(15)+4*IN(16)+2*IN(17)+IN(18)
ISN 0064              IEAD(4)=100*(2*IN(19)+IN(20))+10*(8*IN(21)+4*IN(22)+2*IN(23)+
                     .IN(24))+8*IN(25)+4*IN(26)+2*IN(27)+IN(28)
ISN 0065              IEAD(5)=10*(2*IN(31)+IN(32))+8*IN(33)+4*IN(34)+2*IN(35)+IN(36)
ISN 0066              IEAD(6)=10*(4*IN(37)+2*IN(38)+IN(39))+8*IN(40)+4*IN(41)+2*IN(42)+
                     .IN(43)
ISN 0067              HEAD(7)=10*(4*IN(44)+2*IN(45)+IN(46))+8*IN(47)+4*IN(48)+2*IN(49)+
                     .IN(50)+0.1*(8*IN(51)+4*IN(52)+2*IN(53)+IN(54))
ISN 0068              IEAD(8)=8*IN(57)+4*IN(58)+2*IN(59)+IN(60)
ISN 0069              IEAD(9)=4*IN(64)+2*IN(65)+IN(66)
ISN 0070              HEAD(10)=0.5*IN(67)+IN(68)+2*IN(69)+4*IN(70)
                C
                C        PRINT HEADER IF IPRT=1,2 IN APPROPRIATE FORMAT
                C
ISN 0071              IF (IPRT.EQ.0) GO TO 28
ISN 0073              IF (IPRT.EQ.2) GO TO 65
ISN 0075              IF (IPRT.EQ.4) GO TO 200
ISN 0077              WRITE(6,9) (IEAD(J),J=1,6),HEAD(7),(IEAD(J),J=8,9),HEAD(10)
ISN 0078              IF (IPRT.EQ.3) GO TO 200
ISN 0080              GO TO 28
ISN 0081        65    WRITE(6,3) (IEAD(J),J=1,6),HEAD(7),(IEAD(J),J=8,9),HEAD(10)
```

```
               C
               C         NOW READ DATA: CONVERT<111(5(A1.A2.A3).1(B1.B2.B3))> TO
               C         555A1.555A2.555A3.111B1.111B2.111B3.
               C         MAX. BINARY COUNTS IN ANY CHANNEL IS 36767.
               C         STUFF SEQUENTIAL DIGITAL DATA TO IBUF. SEARCH FOR PARITY BIT AND
               C         LOAD INTO ARRAY IPAR.
               C
ISN 0082          28    IF (IBYTE.EQ.6006) GO TO 31
ISN 0084                DO 32 J=1.1998
ISN 0085          32    IBUF(J)=40000
ISN 0086                GO TO 33
ISN 0087          31    JCNT=1
ISN 0088                DO 26 J=6.3000.3
ISN 0089                DO 22 J1=1, 3
ISN 0090                ICNV(J1*2-1)=IM(J+J1)/256
ISN 0091                IARG=IM(J+J1)
ISN 0092          22    ICNV(J1*2)=MOD(IARG.256)
ISN 0093                DO 24 J1=1.4.3
ISN 0094                IARG=ICNV(J1)
ISN 0095                IL=MOD(IARG.256)
ISN 0096                IR=ICNV(J1+2)/2
ISN 0097                KAP=IL*2048+ICNV(J1+1)*32+IR
ISN 0098                IBUF(JCNT)=LAP(2)
ISN 0099                ICHK=0
ISN 0100                DO 23 J2=1.16
ISN 0101                ICHK=ICHK+MOD(KAP,2)
ISN 0102          23    KAP=KAP/2
ISN 0103                IARG=ICNV(J1+2)
ISN 0104                ICHK=ICHK+MOD(IARG.2)
ISN 0105                IPAR(JCNT)=MOD(ICHK,2)
ISN 0106          24    JCNT=JCNT+1
ISN 0107          26    CONTINUE
               C
               C          CORRECT AND LIST DATA PARITY ERRORS
               C
ISN 0108                DO 29 J=1.1998
ISN 0109                IF (IPAR(J).EQ.1) GO TO 29
ISN 0111                J2=(J-1)/18+1
ISN 0112                J3=J-(J2-1)*18
ISN 0113                J4=(J3-1)/3+1
ISN 0114                J5=J3-(J4-1)*3
ISN 0115                IF (J3.GT.15) GO TO 13
ISN 0117                IF (J5.EQ.1) CH=161
ISN 0119                IF(J5.EQ.2) CH=162
ISN 0121                IF (J5.EQ.3) CH=163
ISN 0123                J7=(J2-1)*5+J4
ISN 0124                GO TO 14
ISN 0125          13    IF (J3.EQ.16) CH=177
ISN 0127                IF (J3.EQ.17) CH=178
ISN 0129                IF (J3.EQ.18) CH=179
ISN 0131                J7=J2*5
ISN 0132          14    WRITE(6.2) CH.J7
ISN 0133                IFOR=3
ISN 0134                IBAK=3
ISN 0135                IF (J4.EQ.5) IFOR=6
ISN 0137                IF (J4.EQ.6) IFOR=18
ISN 0139                IF (J4.EQ.1) IBAK=6
ISN 0141                IF (J4.EQ.6) IBAK=18
ISN 0143                JL=J-IBAK
ISN 0144                JH=J+IFOR
ISN 0145                IF (JL.LT.1) JL=JH
ISN 0147                IF (JH.GT.1998) JH=JL
ISN 0149                IBUF(J)=(IBUF(JL)+IBUF(JH))/2
ISN 0150          29    CONTINUE
ISN 0151          33    CONTINUE
               C
               C          REASSIGN IBUF TO CHANNELS A & B
               C
ISN 0152                JA=0
ISN 0153                JB=0
ISN 0154                JI=1
ISN 0155                DO 39 J=1.111
ISN 0156                DO 34 J1=1.5
ISN 0157                JA=JA+1
ISN 0158                A1(JA)=IBUF(JI)
ISN 0159                A2(JA)=IBUF(JI+1)
ISN 0160                A3(JA)=IBUF(JI+2)
ISN 0161          34    JI=JI+3
ISN 0162                JB=JB+1
ISN 0163            B1(JB)=IBUF(JI)
ISN 0164            B2(JB)=IBUF(JI+1)
ISN 0165            B3(JB)=IBUF(JI+2)
ISN 0166       39   JI=JI+3
              C
              C       NOW PRINT ARRAY IF IPRT=2
              C
ISN 0167            IF (IPRT.NE.2) GO TO 200
ISN 0169            DO 69 J=1.555
ISN 0170            IA1(J)=A1(J)
ISN 0171            IA2(J)=A2(J)
ISN 0172            IA3(J)=A3(J)
ISN 0173            IF (J.GT.111) GO TO 69
ISN 0175            IB1(J)=B1(J)
ISN 0176            IB2(J)=B2(J)
ISN 0177            IB3(J)=B3(J)
ISN 0178       69   CONTINUE
ISN 0179            WRITE(6.7)
ISN 0180            DO 60 L=1,111
ISN 0181            IN(L)=(L-1)*5
ISN 0182            LL=(L-1)*5+1
ISN 0183            LH=LL+4
ISN 0184       60   WRITE(6.4) (IN(L).((IA1(K).IA2(K).IA3(K)).K=LL,LH))
ISN 0185            WRITE(6.8)
ISN 0186            DO 61 L=1.21
ISN 0187            IN(L)=(L-1)*25
ISN 0188            LL=(L-1)*5+1
ISN 0189            LH=LL+4
ISN 0190       61   WRITE(6.4) (IN(L).((IB1(K).IB2(K).IB3(K)).K=LL,LH))
ISN 0191            IN(22)=525
ISN 0192            WRITE(6.6) IN(22).(IB1(111).IB2(111).IB3(111))
ISN 0193            DO 75 J=1.555
ISN 0194            A1(J)=IA1(J)
ISN 0195            A2(J)=IA2(J)
ISN 0196            A3(J)=IA3(J)
ISN 0197            IF (J.GT.111) GO TO 75
ISN 0199            B1(J)=IB1(J)
ISN 0200            B2(J)=IB2(J)
ISN 0201            B3(J)=IB3(J)
ISN 0202       75   CONTINUE
ISN 0203      200   RETURN
ISN 0204            END
```

NAME--ROTATE

TYPE--DATA

SOURCE--D. CHESLEY

PURPOSE--Converts NS and EW data to radial and transverse data. R is positive toward epicenter, T is positive to the left when viewing from station toward the epicenter.

DESCRIPTION--CALL ROTATE (PHI, G, NUM)

PHI - angle of rotation (N through E) in degrees

G - gain of channel A3 relative to A2

NUM - = 0 rotates data returned from FIND
= 1 rotates data returned from READ.

COMMON--/SEARCH/ contains data from FIND to be rotated if NUM = 0.

/SMACK/ contains data from READ to be rotated if NUM = 1

/TURN/ contains rotated data. CHR is radial; CHT is transverse

/ANGLE/ contains label information for PLT1

/X/ contains length of array to be rotated

ROTATE does not change any values in /SEARCH/, /SMACK/, or /X/.

```
ISN 0002                  SUBROUTINE ROTATE(PHI,G,NUM)
                C
                C          ROTATE IS DESIGNED TO ROTATE SEISMOGRAMS IN ORDER
                C          TO CONVERT FROM N-S AND E-W TRACES TO RADIAL AND TRANSVERSE
                C          TRACES.  PHI IS THE ANGLE OF ROTATION IN DEGREES
                C          AND G IS THE GAIN OF THE A2 CHANNEL RELATIVE TO THE A3
                C          CHANNEL.  NUM=0 ROTATES THE CH ARRAYS AS RETURNED FROM
                C          'FIND' AND NUM=1 ROTATES THE A CHANNELS AS RETURNED
                C          FROM 'READ'
                C          WRITTEN BY DUNCAN CHESLEY  HIG-320 X-7070
                C          LATEST VERSION: 7 SEPT 1974
                C
ISN 0003                COMMON/SEARCH/CH1(5550),CH2(5550),CH3(5550)
ISN 0004                COMMON/SMACK/A1(555),A2(555),A3(555),B1(111),B2(111),
                       .B3(111),HEAD(10),IREC
ISN 0005                COMMON/TURN/CHR(5550),CHT(5550)
ISN 0006                COMMON/ANGLE/ANG,GAIN
ISN 0007                COMMON/X/IXMAX
ISN 0008                PHI1=PHI*2.*3.14159265/360.
ISN 0009                ANG=PHI
ISN 0010                GAIN=G
ISN 0011                SINE=SIN(PHI1)
ISN 0012                COSINE=COS(PHI1)
ISN 0013                IIXMAX=IXMAX
ISN 0014                IF (NUM.EQ.1) IIXMAX=555
ISN 0016                DO 100 J=1,IIXMAX
ISN 0017                IF (NUM.EQ.1) GO TO 90
ISN 0019                CHR(J)=G*CH3(J)*SINE+CH2(J)*COSINE
ISN 0020                CHT(J)=-G*CH3(J)*COSINE+CH2(J)*SINE
ISN 0021                GO TO 100
ISN 0022           90   CHR(J)=G*A3(J)*SINE+A2(J)*COSINE
ISN 0023                CHT(J)=-G*A3(J)*COSINE+A2(J)*SINE
ISN 0024          100   CONTINUE
ISN 0025                RETURN
ISN 0026                END
```

NAME--SEC2HD

TYPE--UTILITY

SOURCE--D. CHESLEY

PURPOSE-- converts seconds since 1968 (IGNORING LEAP YEARS and 1-sec adjustments of WWV(H)) to an array of year, day, hour, minute, second.

DESCRIPTION--CALL SEC2HD (IB, IA)

IB - integer number of seconds since 1/1/68

IA - dimension = 5, integer array to contain year, day, hour, minute, second.

NOTES--SEC2HD performs inverse process of HD2SEC.

```
      SUBROUTINE SEC2HD(IB,IA)
      DIMENSION IA(5)
      NUMSEC=365*24*3600
      IA(1)=IB/NUMSEC+1968
      IC=IB-(IA(1)-1968)*NUMSEC
      NUMSEC=24*3600
      IA(2)=IC/NUMSEC
      IC=IC-IA(2)*NUMSEC
      IA(3)=IC/3600
      IC=IC-IA(3)*3600
      IA(4)=IC/60
      IA(5)=IC-IA(4)*60
      RETURN
      END
```

NAME--SETCC, SETNP, SETXYN

TYPE--UTILITY

SOURCE--D. CHESLEY

PURPOSE--Initializes various parameters necessary for correct operation of entire system of subroutines. SETNP is used except in programs that generate a XYNETICS plot or a CALCOMP plot. Those programs use SETXYN and SETCC, respectively.

DESCRIPTION--CALL SETCC, CALL SETNP, CALL SETXYN

SETCC, SETNP or SETXYN must be called once at the beginning of each main program

COMMON/SMACK/ contains IREC, which must be set to zero before any tape reading.

IREC counts records on the data tape /SMACK/ also contains the header array (HEAD(10)), which is initialized to zero

/FILT/ contains label information for PLT1

/PLTPAR/ contains variables used by PLT1

XOFF = x-distance in inches from initial pen position to origin of plot (lower end of Y-axis)

YOFF = Y-distance in inches from initial pen position to origin of plot

H1 = height in inches of small letters in plot

H2 = height in inches of large letters in plot

NOTES--The statement CALL SETCC must be used if the plot is to be made on a CALCOMP plotter. Care must then be taken that the total height of the plot to be generated (including YOFF) does not exceed 11 inches, but no other programming change need be made. The labels for the start time, station, and channel name might not be properly spaced in the CALCOMP plot.

```
ISN 0002            SUBROUTINE SETCC
ISN 0003            COMMON/SMACK/A1(555).A2(555).A3(555).B1(111).B2(111).
                   .B3(111),HEAD(10),IREC
ISN 0004            COMMON/FILT/FILAB(2),SPER,XLPER
ISN 0005            COMMON/PLTPAR/XOFF.YOFF,H1,H2
ISN 0006            DIMENSION IEAD(10),BUFFER(1000)
ISN 0007            EQUIVALENCE(HEAD(1).IEAD(1))
ISN 0008            SPER=9999.
ISN 0009            XLPER=9999.
ISN 0010            IREC=0
ISN 0011            H1=2./25.4
ISN 0012            H2=0.15
ISN 0013            XOFF=1.
ISN 0014            YOFF=0.25
ISN 0015            NBUF=4000
ISN 0016            CALL PLOTS(BUFFER.NBUF)
ISN 0017            DO 25 J=1,10
ISN 0018            IF((J.EQ.7).OR.(J.EQ.10)) GO TO 24
ISN 0020            IEAD(J)=0
ISN 0021            GO TO 25
ISN 0022       24   HEAD(J)=0.
ISN 0023       25   CONTINUE
ISN 0024            RETURN
ISN 0025            END
```

```
ISN 0002            SUBROUTINE SETNP
ISN 0003            COMMON/SMACK/A1(555).A2(555).A3(555).B1(111).B2(111),
                   .B3(111),HEAD(10),IREC
ISN 0004            DIMENSION IEAD(10)
ISN 0005            EQUIVALENCE (HEAD(1).IEAD(1))
ISN 0006            IREC=0
ISN 0007            DO 25 J=1.10
ISN 0008            IF((J.EQ.7).OR.(J.EQ.10)) GO TO 24
ISN 0010            IEAD(J)=0
ISN 0011            GO TO 25
ISN 0012       24   HEAD(J)=0.
ISN 0013       25   CONTINUE
ISN 0014            RETURN
ISN 0015            END
ISN 0002            SUBROUTINE SETXYN
ISN 0003            COMMON/SMACK/A1(555),A2(555),A3(555),B1(111),B2(111),B3(111),
                   .HEAD(10),IREC
ISN 0004            COMMON/FILT/FILAB(2),SPER,XLPER
ISN 0005            COMMON/PLTPAR/XOFF,YOFF,H1,H2
ISN 0006            DIMENSION IEAD(10)
ISN 0007            EQUIVALENCE (HEAD(1),IEAD(1))
ISN 0008            SPER=9999.
ISN 0009            XLPER=9999.
ISN 0010            IREC=0
ISN 0011            H1=2./25.4
ISN 0012            H2=0.15
ISN 0013            XOFF=1.
ISN 0014            YOFF=0.25
ISN 0015            IUNIT=1
ISN 0016            CALL PLOTS(ND,NE,IUNIT)
ISN 0017            DO 25 J=1,10
ISN 0018            IF((J.EQ.7).OR.(J.EQ.10)) GO TO 24
ISN 0020            IEAD(J)=0
ISN 0021            GO TO 25
ISN 0022       24   HEAD(J)=0.
ISN 0023       25   CONTINUE
ISN 0024            RETURN
ISN 0025            END
```

NAME--SHRINK

TYPE-- UTILITY

SOURCE--D. CHESLEY

PURPOSE--makes data returned from FIND correspond exactly to desired start and stop times.

DESCRIPTION--CALL SHRINK (IBEG , IEND)

IBEG - (dimension = 5, integer) contains year, day, hour, minute, and second of desired first data point

IEND - (dimension = 5, integer) contains time of last data point

COMMON--/X/ - IXMAX is the length of the data in CH1, CH2, and CH3. IXMAX is decreased by SHRINK.

/SEARCH/ - contains the data

/SMACK/ - contains HEAD(10), which gives the start time of the data

HEAD(10) is changed by SHRINK

NOTES--READ is capable of reading and interpreting entire records (555 sec). SHRINK takes the data in CH1, 2, and 3, which must be a multiple of the record length, and deletes data from beginning and end so that only the desired data are in CH1, CH2 and 3. SHRINK operates on all three channels simultaneously and is called at the end of FIND.

```
ISN 0002             SUBROUTINE SHRINK(IEEG,IEND)
ISN 0003             COMMON/X/IXMAX
ISN 0004             COMMON/SEARCH/CH1(5550),CH2(5550),CH3(5550)
ISN 0005             COMMON/SMACK/A1(555),A2(555),A3(555),B1(111),B2(111),
                    .B3(111),HEAD(10),IREC
ISN 0006             INTEGER BEG,END,START,STOP
ISN 0007             DIMENSION IEAD(10),IHEAD(5),KSTART(5),IBEG(1),IEND(1)
ISN 0008             EQUIVALENCE(HEAD(1),IEAD(1))
               C
ISN 0009             CALL HD2SEC(IEEG,EEG)
ISN 0010             CALL HD2SEC(IEND,END)
ISN 0011             CALL HDCONV(KSTART,1)
ISN 0012             CALL HD2SEC(KSTART,START)
ISN 0013             STOP=START+IXMAX-1
ISN 0014             KXMAX=END-BEG+1
ISN 0015             DO 20  J=1,KXMAX
ISN 0016             J1=BEG-START+J
ISN 0017             CH1(J)=CH1(J1)
ISN 0018             CH2(J)=CH2(J1)
ISN 0019             CH3(J)=CH3(J1)
ISN 0020          20 CONTINUE
               C
ISN 0021             CALL HDCONV(IEEG,-1)
ISN 0022             IXMAX=KXMAX
ISN 0023             RETURN
ISN 0024             END
```

NAME--TABLE

TYPE--IO

SOURCE--D. CHESLEY

PURPOSE--Prints the output from CORRL8 in a simplified tabular form.

DESCRIPTION--CALL TABLE

COMMON--/PRINT/ contains data to be printed for each channel.

(1) (see below) is calculated from JMAX(I) and JMIN(I)

/COR/ - contains sum trace and necessary start and stop times

/X/ - contains length of sum trace in data points

NOTES-- TABLE prints correlation output for each channel, which includes:

(1) start and stop times for reference and scan,

(2) times of maximum and minimum correlation coefficient,

(3) values of maximum and minimum correlation coefficient,

(4) slopes at maximum and minimum of correlation (relative amplitude of scan with respect to reference),

(5) standard error in slopes,

(6) amplitude of scan signal in digital units (= slope x peak-to-peak amplitude of reference),

(7) standard error of (6).

For the sum trace only (1), (2), and (3) are printed.

```
ISN 0002                SUBROUTINE TABLE
ISN 0003                COMMON/PRINT/JMAX(4),JMIN(4),CMAX(4),CMIN(4),SLOPE1(4),
                       .SLOPE2(4).ERROR1(4).ERROR2(4).AMPL1(4).AMPL2(4).AMPER1(4).
                       .AMPER2(4)
ISN 0004                COMMON/COR/SUM(2700),IREF1(5),IREF2(5),IORIG(5),
                       .ISCAN1(5),ISCAN2(5)
ISN 0005                COMMON/X/IXMAX
ISN 0006                DIMENSION ANAME(4)
ISN 0007                INTEGER HEAD
ISN 0008                DIMENSION HEAD(5),ITIME(5)
ISN 0009                DATA ANAME/'CH 1','CH 2','CH 3','SUM '/
ISN 0010                CALL HDCONV(ITIME,1)
ISN 0011                CALL HD2SEC(ITIME,IT)
ISN 0012                CALL MAXMIN(SUM,IXMAX,CMAX(4),MAX,CMIN(4),MIN)
ISN 0013                JMAX(4)=MAX+IT-1
ISN 0014                JMIN(4)=MIN+IT-1
              C
ISN 0015                PRINT 49
ISN 0016          49    FORMAT(1H1)
ISN 0017                PRINT 20,(IREF1(K),K=1,5),(IREF2(K),K=2,5),(ISCAN1(K),K=1,5),
                       .(ISCAN2(K),K=2,5)
ISN 0018          20    FORMAT(1X,'REF FROM ',I4,'-',I3,'-',2(I2,':'),I2,2X,'TO ',
                       .I3,'-',2(I2,':'),I2,4X,'SCAN FROM ',I4,'-',I3,'-',
                       .2(I2,':'),I2,2X,'TO ',I3,'-',2(I2,':'),I2,//)
ISN 0019                PRINT 9
ISN 0020          9     FORMAT(1X,//,22X,'*****************',//)
ISN 0021                PRINT 10
ISN 0022          10    FORMAT(1X,'CHANNEL',10X,'TIME',8X,'CORR COEF',5X,'SLOPE',
                       .4X,'LOGSLOPE',5X,'ERROR',9X,'AMPLITUDE',3X,'AMP ERROR')
ISN 0023                PRINT 11
ISN 0024          11    FORMAT(1X,//,20X,'DATA FOR MAXIMUM',/)
              C
ISN 0025                DO 100 J=1,3
ISN 0026                CALL SEC2HD(JMAX(J),HEAD)
ISN 0027                IF(SLOPE1(J).EQ.0.) SLOPE=0.
ISN 0029                IF(SLOPE1(J).NE.0.) SLOPE=ALOG10(ABS(SLOPE1(J)))
ISN 0031                PRINT 12,ANAME(J),(HEAD(K),K=1,5),CMAX(J),
                       .SLOPE1(J),SLOPE,ERROR1(J),AMPL1(J),AMPER1(J)
ISN 0032          12    FORMAT (1X,A4,6X,I4,'-',I3,'-',2(I2,':'),I2,3X,F7.4,3X,F11.6,
                       .2X,F8.5,3X,F9.6,4X,F11.2,3X,F9.2,/)
              C
ISN 0033         100    CONTINUE
ISN 0034                CALL SEC2HD(JMAX(4),HEAD)
ISN 0035                PRINT 14,ANAME(4),(HEAD(K),K=1,5),CMAX(4)
ISN 0036          14    FORMAT (1X,A4,6X,I4,'-',I3,'-',2(I2,':'),I2,3X,F7.4,3X,11('+'),
                       .2X,8('+'),3X,9('+'),4X,11('+'),3X,9('+'),/)
              C
ISN 0037                PRINT 13
ISN 0038          13    FORMAT(1X,/,20X,'DATA FOR MINIMUM',/)
ISN 0039                DO 200 J=1,3
ISN 0040                CALL SEC2HD(JMIN(J),HEAD)
ISN 0041                IF(SLOPE2(J).EQ.0.) SLOPE=0.
ISN 0043                IF(SLOPE2(J).NE.0.) SLOPE=ALOG10(ABS(SLOPE2(J)))
ISN 0045                PRINT 12,ANAME(J),(HEAD(K),K=1,5),CMIN(J),SLOPE2(J),
                       .SLOPE,ERROR2(J),AMPL2(J),AMPER2(J)
              C
ISN 0046         200    CONTINUE
ISN 0047                CALL SEC2HD(JMIN(4),HEAD)
ISN 0048                PRINT 14,ANAME(4),(HEAD(K),K=1,5),CMIN(4)
              C
ISN 0049                PRINT 9
ISN 0050                RETURN
ISN 0051                END
```

NAME--WRTSUM

TYPE--IO

SOURCE--D. CHESLEY

PURPOSE--writes the sum trace of one station permanently on disk after several component correlations have been performed. This trace is used for beam focusing.

DESCRIPTION-CALL WRTSUM (IF)

IF - Device or file name in JCL that describes the output file.

COMMON--/SMACK/ contains the start time and station number

/X/ contains the length of the sum trace

/COR/ contains the sum trace

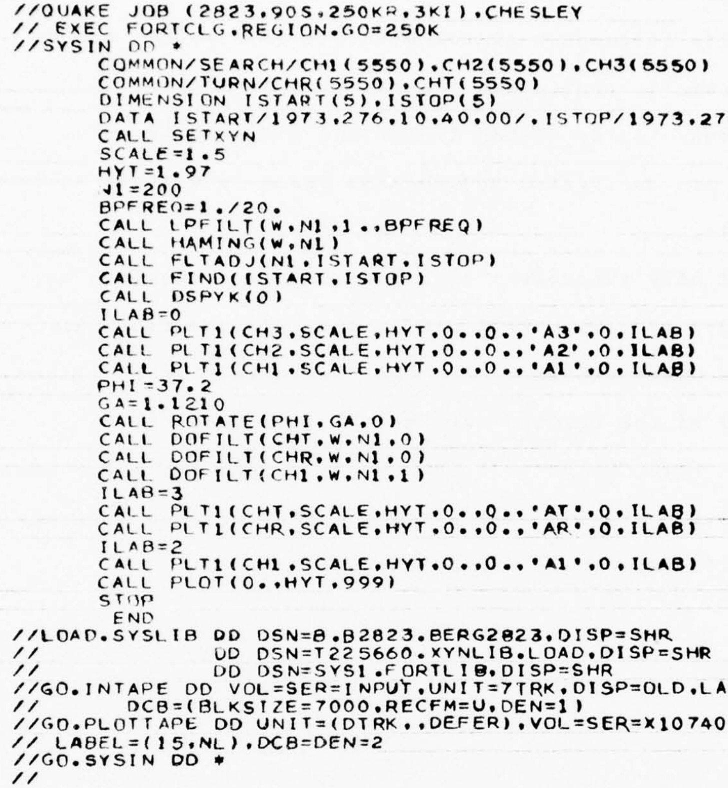
NOTES: User must supply the appropriate JCL with the main program. Refer to sample program CORR for details.

```
ISN 0002          SUBROUTINE WRTSUM(IF)
ISN 0003          COMMON/SMACK/A1(555),A2(555),A3(665),B1(111),
                 ,B2(111),B3(111),HEAD(10),IREC
ISN 0004          COMMON/X/IXMAX
ISN 0005          COMMON/COR/SUM(2700),IREF1(5),IREF2(5),IORIG(5),ISCAN1(5),
                 ,ISCAN2(5)
ISN 0006          DIMENSION IT1(5),IEAD(10)
ISN 0007          EQUIVALENCE (HEAD(1),IEAD(1))
ISN 0008          CALL HDCONV(IT1,1)
ISN 0009          CALL HD2SEC(IT1,IT)
ISN 0010          WRITE(IF) IT,IXMAX,IEAD(2),SUM
ISN 0011          ENDFILE IF
ISN 0012          RETURN
ISN 0013          END
```

SAMPLE PROGRAMS AND JCL

The final section of this report presents three sample programs to show how the subroutines are used. The FORTRAN is simple and will not be discussed. Here we will try only to point out some important JCL considerations.

The first program (QUAKE) shows how the user reads data from the tape, plots the data, and rotates, filters, and plots the results. Note the order of the PLT1 call statements. The final CALL PLOT statement is necessary for all programs that create an output tape for the XYNETICS plotter. The first data set (B.B2823.BERG2823) contains the subroutines described in this report; the second contains the plotting routines that run the XYNETICS plotter (contact KARL HINCK, HIG, for details). The INTAPE DD statement describes the input tape with the data. Any name may be substitued for 'INPUT' but the remainder of the statement must remain as printed. The PLOTTAPE DD statement describes the output tape. X10740 and the file number are the only characters that may be changed. The file number must be 2 or larger.



```
//QUAKE JOB (2823,90S,250KR,3KI).CHESLEY
// EXEC FORTCLG.REGION.GO=250K
//SYSIN DD *
      COMMON/SEARCH/CH1(5550).CH2(5550).CH3(5550)
      COMMON/TURN/CHR(5550).CHT(5550)
      DIMENSION ISTART(5).ISTOP(5)
      DATA ISTART/1973.276.10.40.00/.ISTOP/1973.276.11.15.00/
      CALL SETXYN
      SCALE=1.5
      HYT=1.97
      N1=200
      BPFREQ=1./20.
      CALL LPFILT(W,N1,1.,BPFREQ)
      CALL HAMING(W,N1)
      CALL FLTADJ(N1.ISTART.ISTOP)
      CALL FIND(ISTART.ISTOP)
      CALL DSPYK(0)
      ILAB=0
      CALL PLT1(CH3.SCALE,HYT.0..0.,'A3'.0.ILAB)
      CALL PLT1(CH2.SCALE.HYT.0..0.,'A2'.0.ILAB)
      CALL PLT1(CH1.SCALE.HYT.0..0.,'A1'.0.ILAB)
      PHI=37.2
      GA=1.1210
      CALL ROTATE(PHI.GA.0)
      CALL DOFILT(CHT,W,N1.0)
      CALL DOFILT(CHR,W,N1.0)
      CALL DOFILT(CH1.W,N1.1)
      ILAB=3
      CALL PLT1(CHT.SCALE.HYT.0..0.,'AT'.0.ILAB)
      CALL PLT1(CHR,SCALE,HYT.0..0.,'AR'.0.ILAB)
      ILAB=2
      CALL PLT1(CH1.SCALE,HYT.0..0.,'A1'.0.ILAB)
      CALL PLOT(0..HYT.999)
      STOP
       END
//LOAD.SYSLIB DD DSN=B.B2823.BERG2823.DISP=SHR
//             DD DSN=T225660.XYNLIB.LOAD.DISP=SHR
//             DD DSN=SYS1.FORTLIB.DISP=SHR
//GO.INTAPE DD VOL=SER=INPUT.UNIT=7TRK.DISP=OLD.LABEL=(.NL).
//      DCB=(BLKSIZE=7000.RECFM=U.DEN=1)
//GO.PLOTTAPE DD UNIT=(DTRK..DEFER).VOL=SER=X10740.DISP=OLD.
// LABEL=(15,NL).DCB=DEN=2
//GO.SYSIN DD *
//
```

CORR shows how a correlation is accomplished. In this case a high-pass (in frequency) filter with 401 data points and a cutoff at 45 sec is applied to the data inside COREAD. A rotation is done inside COREAD (THETA and G are non-zero). Two channels are correlated and the sum trace is written in a permanent file.

The BLOCK DATA subprogram contains the times required by COREAD for tape reading. FT11, FT12, and FT13 describe the scratch files where COREAD places the tape data. These files are destroyed at the terminus of the job. FT15 describes the permanent file, which contains the sum trace for later beam focusing. If the program terminates abnormally, step B eleminates this file from the disk and catalog.

```
//CORR  JOB (2823,295KR,3KI,200S),CHESLEY
// EXEC FORTCLG,REGION.LOAD=110K,REGION.GO=295K
//SYSIN DD *
      COMMON/COR/SUM(2700),IREF1(5),IREF2(5),IORIG(5),
     .ISCAN1(5),ISCAN2(5)
      COMMON/ARRAY/N,W(1000)
      CALL SETXYN
      SCALE=1.5
      HYT=1.97
      HPFREQ=1./50.
      N=200
      THETA=24.5
      G=1.
      CALL HPFILT(W,N,1.,HPFREQ)
      CALL HAMING(W,N)
      IL=1
      CALL COREAD(IL,THETA,G)
      LAB=3
      NUMCH=2
      NFILE=15
      CALL CORRL8(2,SCALE,HYT,LAB,NUMCH,NFILE)
      CALL CORRL8(1,SCALE,HYT,LAB,NUMCH,NFILE)
      CALL TABLE
C
      CALL PLT1(SUM,SCALE,HYT,1.,-1.,'SM',0,0)
      CALL PLOT(0.,HYT,999)
      STOP
      END
      BLOCK DATA
      COMMON/COR/SUM(2700),IREF1(5),IREF2(5),IORIG(5),
     .ISCAN1(5),ISCAN2(5)
      DATA IREF1/1973,276,10,40,00/,IREF2/1973,276,10,54,00/
      DATA IORIG/1973,276,10,25,00/
      DATA ISCAN1/1973,276,11,40,00/,ISCAN2/1973,276,12,10,00/
      END
//LOAD.SYSLIB DD DSN=B.B2823.BERG2823,DISP=SHR
//             DD DSN=T225660.XYNLIB.LOAD,DISP=SHR
//             DD DSN=SYS1.FORTLIB,DISP=SHR
//             DD DSN=SYS1.FORTSUB,DISP=SHR
//LOAD.SYSIN DD *
//GO.INTAPE DD VOL=SER=INPUT,UNIT=7TRK,DISP=OLD,LABEL=(7,NL),
//      DCB=(BLKSIZE=7000,RECFM=U,DEN=1)
//GO.PLOTTAPE DD VOL=SER=X10740,UNIT=D7RK,DCB=DEN=2,DISP=OLD,
// LABEL=(25,NL)
//FT11F001 DD UNIT=SYSDA,DCB=(RECFM=VS,LRECL=22204,BLKSIZE=7294,
// BUFNO=1),SPACE=(TRK,(2,1))
//FT12F001 DD UNIT=SYSDA,DCB=(RECFM=VS,LRECL=22204,BLKSIZE=7294,
// BUFNO=1),SPACE=(TRK,(2,1))
//FT13F001 DD UNIT=SYSDA,DCB=(RECFM=VS,LRECL=22204,BLKSIZE=7294,
// BUFNO=1),SPACE=(TRK,(2,1))
//FT15F001 DD UNIT=USERDA,DISP=(,CATLG,DELETE),DSN=B.B2823.TLOCR830,
// DCB=(RECFM=VS,LRECL=22216,BLKSIZE=7294,BUFNO=1),SPACE=(TRK,(2,1))
//FT16F001 DD UNIT=USERDA,DISP=(,CATLG,DELETE),DSN=B.B2823.TLOC1830,
// DCB=(RECFM=VS,LRECL=22216,BLKSIZE=7294,BUFNO=1),SPACE=(TRK,(2,1))
//GO.SYSIN DD *
//B EXEC PGM=IEHPROGM,COND=ONLY
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
 UNCATLG DSNAME=B.B2823.TLOCR830
 UNCATLG DSNAME=B.B2823.TLOC1830
//
```

SUMAVE shows how the permanent files created by several runnings of CORR may be added for a beam focusing effect. This program averages the part of each sum trace which is in common with the other sum traces, plotting the original and the averaged sum traces.

```
//SUMAVE JOB (2823.40S.295KR.3KI).CHESLEY
// EXEC FORTCLG.REGION.GO=295K
//SYSIN DD *
       COMMON/X/IXMAX
       COMMON/SMACK/A1(555),A2(555),A3(555),B1(111),
      .B2(111),B3(111),HEAD(10),IREC
       DIMENSION IEAD(10),XSUM(2700,10),IXĈ(10),ITIME(10),ASUM(2700),
      .IHEAD(5),JHEAD(5),ION(10),IOFF(10)
       EQUIVALENCE (HEAD(1),IEAD(1))
       CALL SETXYN
       IBEG=-1E+10
       IEND=-IBEG
       SCALE=1.5
       HYT=1.97
       IPLT=1
       NUMSUM=4
       IB=0
       IDEV=10+NUMSUM
       DO 100 I=11,IDEV
       IB=IB+1
       READ(I) ITIME(IB),IXC(IB),IEAD(2),ASUM
       IF(IPLT-1)80,50,80
 50    CALL SEC2HD(ITIME(IB),IHEAD)
       CALL HDCONV(IHEAD,-1)
       IXMAX=IXC(IB)
       CALL PLT1(ASUM,SCALE,HYT,1.,-1.,'SM',0,0)
 80    J2=IXC(IB)
       DO 90 J=1,J2
 90    XSUM(J,IB)=ASUM(J)
 100   CONTINUE
C
C           CALCULATE START AND STOP TIMES
C
       DO 200 I=1,NUMSUM
       IBEG=MAX0(IBEG,ITIME(I))
       IS=ITIME(I)+IXC(I)-1
       IEND=MIN0(IEND,IS)
 200   CONTINUE
C
C         CALC. OFFSETS FOR EACH TRACE
C
       DO 300 I=1,NUMSUM
       ION(I)=IBEG-ITIME(I)+1
       IOFF(I)=IEND-IBEG+ION(I)
 300   CONTINUE
       DO 325 K=1,2700
 325   ASUM(K)=0.
C
C         AVERAGE THE SUMS
C
       DO 400 I=1,NUMSUM
       K=0
       J1=ION(I)
       J2=IOFF(I)
 341   CONTINUE
       DO 350 J=J1,J2
       K=K+1
       ASUM(K)=ASUM(K)+XSUM(J,I)
 350   CONTINUE
 400   CONTINUE
       DO 450 J=1.K
 450   ASUM(J)=ASUM(J)/NUMSUM
C
C          PLOT THE AVERAGE
C
       IXMAX=K
       CALL SEC2HD(IBEG.IHEAD)
       CALL HDCONV(IHEAD,-1)
       IEAD(2)=0
       CALL PLT1(ASUM,SCALE,HYT,+1.,-1.,'AV',0,0)
C
C         PRINT RESULTS
C
       CALL MAXMIN(ASUM,IXMAX,SMAX,JMAX,SMIN,JMIN)
       JMAX=JMAX+IBEG-1
       JMIN=JMIN+IBEG-1
       CALL SEC2HD(JMAX,IHEAD)
       CALL SEC2HD(JMIN,JHEAD)
       PRINT 402,(IHEAD(K),K=1,5),SMAX
       PRINT 403,(JHEAD(K),K=1,5),SMIN
 402   FORMAT(1X,'TIME OF MAX: ',I4,'-',I3,2X,I2,':',I2,':',I2,'  MAX C'
      .,'OEFF: ',F6.3,//)
 403   FORMAT(1X,'TIME OF MIN: ',I4,'-',I3,2X,I2,':',I2,':',I2,'  MIN C'
      .,'OEFF: ',F6.3,//)
       CALL PLOT(0.,HYT,999)
       STOP
       END
//LOAD.SYSLIB DD DSN=B.B2823.BERG2823,DISP=SHR
//             DD DSN=T225660.XYNLIB.LOAD,DISP=SHR
//             DD DSN=SYS1.FORTLIB,DISP=SHR
//             DD DSN=SYS1.FORTSUB,DISP=SHR
//LOAD.SYSIN DD *
//GO.PLOTTAPE DD VOL=SER=X51698,UNIT=DTRK,DCB=DEN=2,DISP=OLD,
// LABEL=(11,NL)
//FT11F001 DD DSN=B.B2823.KIPF,DISP=OLD
//FT12F001 DD DSN=B.B2823.MATF,DISP=OLD
//FT13F001 DD DSN=B.B2823.TLOFR,DISP=OLD
//FT14F001 DD DSN=B.B2823.CHGF,DISP=OLD
//GO.SYSIN DD *
//
```

FILTER1 illustrates the use of INFILT to retrieve the ground motion by deconvolving the instrument response from the tape data. This program calculates and plots a 'filter' for each channel. It reads the tape and applies these 'filters' and then plots the ground motion. Finally, FILTER1 filters the data with a high-pass filter and plots the results. The data set B.B2823.INRESP contains the Fourier coefficients of the instrument response for each of the three channels.

```
//FILTER1 JOB (2823,2805,295KR,4KI).CHESLEY
// EXEC FORTCLG,REGION.GO=295K
//SYSIN DD *
      COMMON/SMACK/ A1(555).A2(555).A3(555).B1(111).
     .B2(111),B3(111),HEAD(10),IREC
      COMMON/SEARCH/CH1( 5550).CH2(5550).CH3(5550)
      COMMON/X/IXMAX
      DIMENSION ISTART(5).ISTOP(5),W(1000)
      DIMENSION W1(513),W2(513).W3(513)
      DATA ISTART/1973,276.10,40,00/, ISTOP/1973,276.11,20.00/
      CALL SETXYN
      SCALE=1.5
      HYT=1.97
      N=256
      N1=400
      NSUM=N+N1
      CALL FLTADJ(NSUM.ISTART.ISTOP)
      HPFREQ=1./100.
      CALL HPFILT(W.N1,1..HPFREQ)
      CALL HAMING(W.N1)
      ILAB=0
      IXMAX=513
      CALL INFILT(1,W1,N,1,14)
      CALL INFILT(2.W2,N,1.14)
      CALL INFILT(3,W3,N,1,14)
      CALL PLT1(W3.SCALE,HYT.0..0..'F3',0.ILAB)
      CALL PLT1(W2.SCALE,HYT,0..0..'F2',0,ILAB)
      CALL PLT1(W1,SCALE,HYT,0..0..'F1',0.ILAB)
      CALL FIND(ISTART,ISTOP)
      CALL DSPYK(0)
      CALL DOFILT(CH3.W3.N.0)
      CALL DOFILT(CH2.W2,N,0)
      CALL DOFILT(CH1.W1.N.1)
      CALL PLT1(CH3.SCALE,HYT,0..0..'G3',0.ILAB)
      CALL PLT1(CH2.SCALE.HYT.0..0..'G2',0.ILAB)
      CALL PLT1(CH1.SCALE,HYT,0..0..'G1',0.ILAB)
      CALL DOFILT(CH3.W.N1.0)
      CALL DOFILT(CH2.W.N1.0)
      CALL DOFILT(CH1.W.N1,1)
      CALL PLT1(CH3.SCALE,HYT,0..0..'G3',0.2)
      CALL PLT1(CH2.SCALE,HYT,0..0..'G2',0.2)
      CALL PLT1(CH1.SCALE,HYT.0..0..'G1',0.2)
      CALL PLOT(0..HYT.999)
      STOP
      END
//LOAD.SYSLIB DD DSN=B.B2823.BERG2823.DISP=SHR
//              DD DSN=T225660.XYNLIB.LOAD,DISP=SHR
//              DD DSN=SYS1.FORTLIB,DISP=SHR
//              DD DSN=SYS1.FORTSUB,DISP=SHR
//LOAD.SYSIN DD *
//GO.INTAPE DD UNIT=7TRK.VOL=SER=X51045.DISP=OLD.
//   DCB=(RECFM=U.BLKSIZE=7000.DEN=1).LABEL=( .NL)
//GO.PLOTTAPE DD VOL=SER=X51698.DISP=OLD,DCB=DEN=2,UNIT=DTRK.
// LABEL=(22.NL)
//GO.FT14F001 DD DSN=B.B2823.INRESP,DISP=SHR
//GO.SYSIN DD *
//
```